AFRL-RY-WP-TR-2011-1022

A SHEAF-THEORETIC/BAYESIAN APPROACH FOR SHAPE REPRESENTATION

Kirk Sturtz

Universal Mathematics

OCTOBER 2010
Final Report

Approved for public release; distribution unlimited.

See additional restrictions described on inside pages

STINFO COPY

AIR FORCE RESEARCH LABORATORY
SENSORS DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7320
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the USAF 88th Air Base Wing (88 ABW) Public Affairs Office (PAO) and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RY-WP-TR-2011-1022 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

*//Signature//

JARED CULBERTSON, Program Manager
Electro Optic Exploitation Technology Branch
Sensor ATR Technology Division

//Signature//

CLARE MIKULA, Branch Chief
Electro Optic Exploitation Technology Branch
Sensor ATR Technology Division

//Signature//

CHRIS RISTICH, Division Chief
Sensor ATR Technology Division
Sensors Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

*Disseminated copies will show "//Signature//" stamped or typed above the signature blocks.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE ***(DD-MM-YY)*** October 2010	**2. REPORT TYPE** Final	**3. DATES COVERED** ***(From - To)*** 10 March 2005 – 31 October 2010

4. TITLE AND SUBTITLE
A SHEAF-THEORETIC/BAYESIAN APPROACH FOR SHAPE REPRESENTATION

5a. CONTRACT NUMBER
FA8650-05-1-1801

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
62204F

6. AUTHOR(S)
Kirk Sturtz

5d. PROJECT NUMBER
6095

5e. TASK NUMBER
04

5f. WORK UNIT NUMBER
60950417

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Universal Mathematics
3274 S. Union Road
Dayton, OH 45418-1331

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory
Sensors Directorate
Wright-Patterson Air Force Base, OH 45433-7320
Air Force Materiel Command
United States Air Force

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
AFRL/RYAT

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-RY-WP-TR-2011-1022

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES
PAO Case Number: 88ABW-2011-0387; Clearance Date: 28 Jan 2011. This report contains color.

14. ABSTRACT

A point to surface matching algorithm for purposes of matching 3D point cloud data to CAD models is developed. By discretization of the unknown transformation parameters and using assignment correspondence modeling the resulting objective function is a multilinear programming problem with decoupled linear constraints. Mathematica software for the optimization algorithm and model setup is provided. The inputs for this code are the CAD models and point cloud 3D data.

15. SUBJECT TERMS
automatic target recognition, object image matching, point to surface matching

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT:**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** (Monitor)
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	SAR	76	Jared Culbertson **19b. TELEPHONE NUMBER** *(Include Area Code)* N/A

Final Report:

March 10, 2005 - October 31, 2010

A Sheaf-Theoretic/Bayesian Approach for Shape Representation

Grant FA8650-05-1-1801

Universal Mathematics

October 31, 2010

Technical Report

Executive Summary

For geometric based object recognition problems the current best approach is to use a “Point to Surface”, or more generically speaking a “Point to Complex Model”, matching algorithm which we have developed over the last five years of this program. This approach is very general addressing many recognition problems beyond simple target recognition problems with their standard CAD model representation. A CAD model representation of an object is a special instance of a *complex* model representation. A complex is a mathematical structure composed of simplices. A point is a 0-simplex, a line a 1-simplex, a triangle a 2-simplex, a tetrahedron a 3-simplex, and so forth.

In most, if not all, target recognition problems the Complex Model approach provides a representation of the object as a composite (or gluing together) of many separate parts (generally surfaces or lines). This model is then to be matched against given image data. By using barycentric coordinates to represent points on each simplex the point to point matching problem can be avoided (or at least greatly alleviated). This correspondence problem has plagued the recognition community for too long because we have viewed it strictly as a discrete combinatorial problem. By embedding the discrete combinatorial problem into the reals this issue is more effectively solved.

This research actually goes beyond the specific representation of the object. To understand this note that there are really three components of any matching problem that can and should be considered separately:

1. the object and image representation,
2. the generic optimization model involving the representations, and
3. the algorithm employed to solve the model.

The recognition of this decomposition of the problem should not be underestimated as parts are interchangeable. To emphasize this aspect we have addressed the solution of the Rubiks Cube problem. For the Rubiks Cube problem the object and image use a simple (non Complex Model) representation in terms of corners, edges, and orientations. The specifics can be found in the DASP 2009 and SPIE 2009 papers. But in both problems, target recognition and Rubiks Cube, the generic optimization model and algorithm used to solve the model arei identical. It is essentially a Least Squares Model using the Frobenious norm which is equivalent to "maximal correspondence" problem. This optimization model is a multilinear programming problem with a multilinear objective function and decoupled linear constraints. Because of its special structure it can be solved relatively efficiently (compared with past methods).

The biggest shortcoming of our research, and the current state of the art in recognition, is that we have not been able to develop a convex optimization model and as such there is the pitfall that the optimization algorithm can return a local minimum rather than a global minimum. For target recognition problems this shortcoming does not have the same impact as the more general problem like Rubiks Cube because further data can be employed to improve the probability of obtaining a global match. (Man made objects tend to have further structure which can be exploited for improving the probability of correct recognition.) We feel strongly that any further progress into recognition must address this local/global issue by a paradigm shift - moving away from a least squares based model (or any of its equivalent representations such as the maximal correspondence model). Any nonconvex optimization model will be plagued by this issue; there is no algorithm that can quarantee a global minimum for such models. Here we note that some authors claim to have made significant progress on this global optimization. In our review of the literature nothing could be further from the truth. Methods that rely on a branch and bound procedure (or box and bound, octree's etc.) all rely on the model being matching n points to n points where the translation component can be easily eliminated. Then by using Lipshitz Theory can use discretization of the unknown rotations and put bounds on how much the objective function can change in a small box. This effectively eliminates the rotations variables (by discretizing over them and searching over all possibilities). At this point the only unknown is the assignment correspondence variables so the model is a simple linear programming problem which has a global minimum (though the argument of the minimum need not be unique). This whole approach collapses when one is trying to match n points with m points (or m points to n surfaces). The translation variables cannot be removed and the problem is 4-linear which does not have a unique local minimum (hence global minimum).

An understanding of the theory underlying the model suggest a relationship to fundamental ideas of quantum mechanics - for example the idea of a superposition of possibilities which the model employs. We note that this is really just an *interpretation* with respect to the variables which can be interpretted as probabilities[1]. However this analogy should not be disregarded; Shroedingers equation, which is a model for the position of a particle, has many solutions depending upon the spectrum of the operator (for the specific problem model). These many solutions (or orbits when considering an electrons position) can be analogized with the many local minimum of a recognition problem. The most probable solution corresponds to a minimuml energy state which would be a "global solution". Any such paradigm shift based upon this analogy would replace the least squares optimization model by an expression modeling the problem - such as an equation rather than an objective function. An optimization algorithm would then not be relevant to the solution but rather other

[1] This relates to the reason for the development of the Category of Probabilistic Mappings by William Lawvere back in the 1960's. He recognized that the deterministic world (based upon aristetelian logic) was a special case of a more general logic. Fifty years later Lawvere is still working on this problem and attempting to axiomatize physics based upon categories more general than the category of sets. In particular, a topos can replace the traditional sets upon which so much of traditional mathematics and science is based. A main interest of his is mechanics and his 1963 seminar paper on Probabilistic Mappings references the issues of interest which relate to recognition. Indeed to talks directly about the recognition problem in terms of probabilistic mappings which lead to our development and understanding.

(numerical) methods would come to the forefront. To date our very limited attempts at such an approach using stochastic matrices rather than permutation (deterministic) matrices have failed because of lack of knowledge on how to choose an appropriate basis. This novel approach is reported on since it generalizes our current method and in our opinion holds the best probability of future progress. Furthermore it allows for the replacement of the optimization model by an equation.[2]

Outline

We break the report down into applications and theory. For someone wanting to know how to use these ideas knowledge of the theory is not necessary; only the representation and modeling aspects are important. On the other hand, the theory provides both the motivation, underpinnings, and directions for future research (or at least indicates what avenues have already been explored).

Our main application is to recognition of targets - those objects which can be represented by a complex. There are many offshoots and add ons which can be exploited for this specific type of problem. For example one can generalize the model by using an Image Point Spread Model which allows for uncertainty in the image data, or one can weight the image points representing the fact that we may have confidence in some points more than others. There are many such generalizations to the core problem and we enumerate them with their models.

There are two approaches to addressing the theory. The motivational approach - how we can to this methodology based upon basic ideas from category theory which view deterministic mappings as a left adjunct of probabilistic mappings and allows for further sheaf theoretic description, or the "simplified"

[2] We recognize that an equation can be viewed as a particular instance of an optimization model where the error (objective function) assumes a value of zero. However there is often a change of emphasis on the technique used to solve an equation rather than an optimization problem.

approach which distills the problem down to a basic least squares problem. We address both of these approaches. The sheaf theoretic approach also describes the Beck-Chevalley Theorem applied to measures which is a nice application.

0.1 3D-3D Ladar

The *3D-3D* registration problem (also known as *3D-3D* alignment) is a fundamental problem in computer vision. In this problem two sets of 3D points are given and the task is to optimally align these two sets of points by determining a group transformation (an affine, euclidean, projective or other transformations depending upon the camera model) so as to minimize the mean squared error between the two sets of points. The two sets can be of different sizes, say of sizes m and n with $m < n$ in which case the subproblem of selecting m points from the set of n points needs to be determined before the alignment itself can be determined. Because this problem involves two sets of points it is also referred to as point to point matching.

Point to surface matching generalizes point to point matching and permits the correspondence between each point composing a *point cloud of image data* with a point on a surface of a CAD model (or other idealized model). The combinatorial problem of matching m points to n points now becomes matching each of the m points in the point cloud to a correspondence point on one of the n surfaces. By introducing assignment (correspondence) parameters that associate with each image point i the "probability" of matching with surface component j, the discrete matching problem can be embedded into the continuous domain and the combinatorial explosion that occurs with matching can be significantly alleviated from the computational perspective. Indeed, for a fixed group transformation this matching problem is a linear programming problem for point to point matching and a quadratic programming problem for point to surface matching. Conversely, for known correspondences (point to point or point to surface) the problem is a *pure* pose estimation problem. For this reason the registration problem is also called the simultaneous pose and correspondence problem.

The challenge in solving this problem is to determine the global optimal solution. Even for the point to point matching problem this is not an easy task. The most popular methods for solving this problem are based upon an alternating minimization procedure between solving the pose problem and the correspondence problem. These methods are all *local* methods as no global optimum is guaranteed. The method based upon Lipschitz optimization[3] has limited use since it is restricted to matching n points to n points.

The approach discussed here is based upon the discretization of the transformation parameters as well so that the problem becomes a multilinear programming problem in the correspondence and pose (transformation group) parameters. The theoretical basis of this model begins with abstracting the general recognition problem as one of a least squares problem.

0.1.1 Least Squares Model

Define a Least Squares Error Recognition Problem as one which can be modeled as the determination involving the Frobenius norm minimization of the error between an Object $\mathcal{O}$, an image $\mathcal{I}$, and a transformation group $\mathcal{G}$ which acts on the image,

$$\min_{g \in \mathcal{G}} ||\mathcal{O} - g\mathcal{I}||_F^2. \tag{1}$$

This model involves the determination of the argument of the minimum (argmin) of the minimuml objective function value which most closely matches the object with the transformed image. The objective function is a $\mathcal{G}$-invariant (pseudo) metric measuring the "distance" between the object and the transformed image. Many recognition problems can be formulated in this fashion, and a linear representation of the object, image, and transformation group are applicable. Such is the case for 3D Euclidean recognition problems where the object and image can be given a matrix representation where the columns (or

[3] The 3D Registration Problem Revisited, Hongdong Li and Richard Hartley, ICCV 2007.

rows) of the matrices correspond to ordered image points and points on the object are given by barycentric coordinates on the surface components.

Using assignment variables allows one to consider unordered collections of points. To each image point indexed by i there corresponds an object point indexed by j. So we associate a variable $p_{i,j}$ which assumes the value 1 provided that image point i corresponds to the object point j and $p_{i,j} = 0$ otherwise. Consider the 2D point to point matching problem. Here the optimization model for matching unordered points $\{u_i\}_i$ to object points $\{x_j\}_j$ is

$$\begin{array}{ll} \min\limits_{R \in SO(2), T \in \Re^2, p_{i,j}} & \sum_i \sum_j p_{i,j} ||(Ru_i + T) - x_j||^2 \\ \text{subject to} & \\ \quad \sum_j p_{i,j} = 1 \quad \forall i & \\ \quad p_{i,j} \geq 0 \quad \forall i, j & \end{array} \tag{2}$$

By the introduction of barycentric coordinates associated with the object points $\{x_j\}$ this problem can be extended to matching image points to lines, surfaces, or any convex hull of a collection of points by the optimization model

$$\begin{array}{ll} \min\limits_{R \in SO(2), T \in \Re^2, p_{i,j}, \alpha_{i,j,k}} & \sum_i \sum_j p_{i,j} ||(Ru_i + T) - \sum_k \alpha_{i,j,k} x_j^k||^2 \\ \text{subject to} & \\ \quad \sum_j p_{i,j} = 1 \quad \forall i & \\ \quad p_{i,j} \geq 0 \quad \forall i, j & \\ \quad \sum_k \alpha_{i,j,k} = 1 \quad \forall i, j & \\ \quad \alpha_{i,j,k} \geq 0 \quad \forall i, j, k & \end{array} \tag{3}$$

where the $\alpha_{i,j,k}$ are the barycentric coordinate associated with the assignment of image point u_i to surface j which has vertices $\{x_j^1, \ldots, x_j^{N_{v_j}}\}$ where N_{v_j} is the number of vertices characterizing the convex hull defining surface j.

The challenge with this model, as with all current models that we are aware of[4], is that because the model is not a convex optimization problem (convex

[4] If one restricts the problem to matching m points to m points then the problem is significantly simpler because the translation variables can be removed from the problem. The paper [**?**] uses Lipschitz global optimization theory to address matching m points to m points.

objective function with convex constraints) the algorithmic solution to this problem cannot guarantee a global minimum. The ability to obtain a global minimum can be significantly improved by taking a "probabilistic perspective" of the problem and exploiting additional information (or extracting information from the given data). Towards this end we discretize the rotation and translation parameters.

0.1.2 Discretization of the Transformation Group Parameters.

In the model (1) suppose $\mathcal{G}$ is a finite discrete transformation group which can be written as a composition of subgroupoids,[5] $\mathcal{G} = \mathcal{G}^1 \dots \mathcal{G}^L$. If $k_\ell = |\mathcal{G}^\ell|$, then for each ℓ, fix a labelling $\{g_1^\ell, g_2^\ell \dots, g_{k_\ell}^\ell\}$ of $\mathcal{G}^\ell$. Let $J = \{1, \dots, k_1\} \times \dots \times \{1, \dots, k_L\}$ be an index set so that any $\mathbf{j} \in J$ corresponds to an element $g_{\mathbf{j}} = g_{j_1}^1 \cdots g_{j_L}^L$ in $\mathcal{G}$.[6] The set of *all possible values* the objective function $||\mathcal{O} - g\mathcal{I}||_F^2$ can assume is

$$\{e_{\mathbf{j}} \doteq ||\mathcal{O} - g_{\mathbf{j}}\mathcal{I}||_F^2\}_{\mathbf{j} \in J}. \tag{4}$$

Associate with each error term $e_{\mathbf{j}}$ a weight $p_{\mathbf{j}} \doteq p_{j_1}^1 \cdots p_{j_2}^2 p_{j_L}^L$ and consider the weighted sum of all these error components

$$\sum_{\mathbf{j} \in J} p_{\mathbf{j}} e_{\mathbf{j}}. \tag{5}$$

Since the index $\mathbf{j}$ specifies the components of the element $g_{\mathbf{j}} = g_{j_1}^1 \cdots g_{j_L}^L$ associated with the error $e_{\mathbf{j}}$ there is a bijective correspondence between the components $g_{j_\ell}^\ell$ and $p_{j_\ell}^\ell$. Restrict the variables $p_{j_\ell}^\ell$ to satisfy $\sum_{i=1}^{k_\ell} p_i^\ell = 1$ and $p_i^\ell \geq 0$ for all components ℓ and all element indices i. For each $\mathcal{G}^\ell$, the set $\{g_1^\ell, g_2^\ell \dots, g_{k_\ell}^\ell\}$ is a finite sample space and each vector $(p_1^\ell, p_2^\ell, \dots, p_{k_\ell}^\ell)$ associated with that component can be interpreted as a distribution on the (global) ℓ^{th} component of an optimal transformation $\hat{g}_{\mathbf{j}}$ minimizing the expression (5). Thus for $\mathcal{G}$ a

[5] A subgroupoid in this setting means a subset of $\mathcal{G}$ which contains the identity of $\mathcal{G}$ and inverses for each element, but the binary operation of $\mathcal{G}$ is restricted to pairs whose product is in the subset (the operation inherits associativity, of course, when defined).

[6] The decomposition of elements $\mathbf{g} \in \mathcal{G}$ need not be unique. One can have $j, i \in J$ with $\mathbf{g_j} = \mathbf{g_i}$ and $j \neq i$.

discrete group, (1) is equivalent to the optimization model

$$\begin{gathered}\min_{p_i^\ell} \sum_{\mathbf{j}\in J} p_{\mathbf{j}} e_{\mathbf{j}} \\ \text{subject to} \\ \sum_{i}^{k_\ell} p_i^\ell = 1 \text{ (for all } \ell) \qquad \text{and} \qquad p_i^\ell \geq 0 \text{ (for all } \ell \text{ and all } i)\end{gathered} \tag{6}$$

By the form of the objective function a global minimum can always be obtained corresponding to a dirac probability measure at each component - *i.e.* if $e_{\mathbf{j}_0}$ is minimuml over all $\mathbf{j} \in J$, then setting the components of its coefficient to one and all the others to zero yields a global minimum.[7] Such a determination of variables will satisfy the constraints and is a dirac measure.

For 3D recognition problems the Euclidean group can be decomposed into six linear transformation groups involving the rotation and translation groups, $\{T_x, T_y, T_z, R_x, R_y, R_z\}$. By discretizing each group, an approximation to the least squares problem (1) model in the form of (6) can be obtained.

0.1.3 The Discretized Least Squares Model with Assignment Correspondence

The 2D point to surface model with discretized transformation group parameters can be written as

$$\begin{aligned} &\min_{p^R_\alpha, p^x_\beta, p^y_\gamma, p^a_{i,j}, \alpha_{i,j,k}} \quad \sum_i \sum_j \sum_\alpha \sum_\beta \sum_\gamma p_{i,j} p^R_\alpha p^x_\beta p^y_\gamma ||(R_\alpha u_i + T_{\beta,\gamma}) - \textstyle\sum_k \alpha_{i,j,k} x_j^k||^2 \\ &\text{subject to} \\ &\quad \textstyle\sum_j p^a_{i,j} = 1 \quad \forall i \quad \sum_j p^R_\alpha = 1 \qquad \sum_\beta p^x_\beta = 1 \qquad \sum_\gamma p^y_\gamma = 1 \\ &\quad p^a_{i,j} \geq 0 \quad \forall i,j \qquad p^R_\alpha \geq 0 \quad \forall \alpha \quad p^x_\beta \geq 0 \quad \forall \beta \quad p^y_\gamma \geq 0 \quad \forall \gamma \\ &\quad \textstyle\sum_k \alpha_{i,j,k} = 1 \quad \forall i,j \\ &\quad \alpha_{i,j,k} \geq 0 \forall i,j,k \end{aligned} \tag{7}$$

where $\{R_\alpha\}_\alpha$ and $\{T_{\beta,\gamma}\}_{\beta,\gamma} = \left\{ \begin{array}{c} \overline{x}_\beta \\ \overline{y}_\gamma \end{array} \right\}_{\beta,\gamma}$ are both a set of constant values.

[7] The minimum objective function value may not have a unique argmin.

The computational effort associated with this model is greatly simplified by expanding the L_2 norm in terms of the inner product, $||x||^2 = \langle x, x \rangle$, and bringing the summation terms inside wherever possible. By exploiting the fact that the variables satisfy $\sum_\alpha p_\alpha^R = 1$, etc., and centering the image data the above objective function simplifies to

$$\begin{aligned}
&\sum_{i=1}^{N_i} \langle u_i, u_i \rangle - 2 \sum_{i=1}^{N_i} \langle (\sum_\alpha p_\alpha^R R_\alpha) u_i, \sum_j \sum_k p_{i,j}^a \alpha_{i,j,k} x_j^k \rangle \\
&-2 \langle \left\{ \begin{array}{c} \sum_\beta p_\beta^x \overline{x}_\beta \\ \sum_\gamma p_\gamma^y \overline{y}_\gamma \end{array} \right\}, \sum_{i=1}^{N_i} \sum_j \sum_k p_{i,j}^a \alpha_{i,j,k} x_j^k \rangle + N_i \sum_{p_\beta^x} \sum_{p_\gamma^y} p_\beta^x p_\gamma^y (\overline{x}_\beta^2 + \overline{y}_\gamma^2) \qquad (8) \\
&+ \sum_{i=1}^{N_i} \sum_j p_{i,j}^a \langle \textstyle\sum_k \alpha_{i,j,k} x_j^k, \sum_{k'} \alpha_{i,j,k'} x_j^{k'} \rangle
\end{aligned}$$

The computation of the barycentric coordinates is where all the computation time is required. As the distributions for each image point u_i converge, $p_{i,j}^a \rightarrow \delta$ so the distribution has components with value 0, the computations become simple. Furthermore the decoupled constraints makes the computation for the projection of the gradients onto the nullspace very efficient for an optimization algorithm. The extension of this model to 3D matching is straightforward. Some typical 3D results are shown in section 0.1.6.

0.1.4 The Maximal Correspondence Model

Going back to the notation of model (1) and it's discretization in section 0.1.1 consider the notation $\langle A, B \rangle_F \doteq Tr(AB^T)$. By using the normalization constraints on the set of variables $\{p_i^\ell\}_i$, and exploiting the bilinearity of the operator $\langle \cdot, \cdot \rangle_F$ the objective function can be written as

$$\begin{aligned}
&\frac{1}{2} ||\mathcal{O}||_F^2 + \frac{1}{2} ||\mathcal{I}||_F^2 - \sum_{\mathbf{j} \in J} p_{\mathbf{j}} \langle \mathcal{O}, g_{\mathbf{j}} \mathcal{I} \rangle_F \\
&= constant - \langle \mathcal{O}, \sum_{\mathbf{j} \in J} p_{\mathbf{j}} g_{\mathbf{j}} \mathcal{I} \rangle_F .
\end{aligned} \qquad (9)$$

Defining the *superposition of transformations* as $P^\ell = \sum_{i=1} p_i^\ell g_i^\ell$, dropping the constant term, and using $\min -z = \max z$ yields the model

$$\max_{p_i^\ell} \quad \langle \mathcal{O}, \prod_{\ell=1}^{L} P^\ell \mathcal{I} \rangle_F^2 \tag{10}$$

subject to the same linear constraints in ((6)). This model is called the *Maximal Correspondence Model* and is the basis of the Map Seeking Algorithm[8]. In their models they also discretize space rather than using barycentric coordinates.[9] Hence the minimuml error and maximal correspondence formulations are equivalent.

0.1.5 An Optimization Algorithm for the MultiLinear Model

Using the notation of section 0.1.1 let $\mathbf{p} = \oplus_{\ell=1}^{L} \mathbf{p}^\ell$ be the direct sum of the L probability distributions $\mathbf{p}^\ell = (p_1^\ell, \ldots, p_{k_\ell}^\ell)$. For the 2D matching problem above, one may have $\mathbf{p}^1 = \{p_1^x, \ldots, p_{N_x}^x\}$, the distribution associated with the optimal x-translation. Starting with an initial uniform distribution for each $\mathbf{p}^\ell$, the decoupled constraints permit the updating of the probability distributions associated with each unknown parameter in parallel which, depending upon the number of processors used, is considerably more efficient than a sequential procedure. Using projected gradients the search direction can be computed as $\mathbf{d} = \oplus_{\ell=1}^{L} \mathbf{d}^\ell$ and each component of the direct sum $\oplus_{\ell=1}^{L} \mathbf{p}^\ell$ can be simultaneously updated via a step $\mathbf{p}^\ell = \mathbf{p}^\ell + \alpha \mathbf{d}^\ell$. A line search procedure can be used to calculate a step length α or, upon normalization of the search directions $\mathbf{d}^\ell$ in each subspace, a constant step length α at each iteration can be employed. Our experience suggest the constant step length is just as effective as a line search procedure and computationally more efficient.

[8] David Arathorn. *Map Seeking Circuits in Visual Cognition: A Computational Mechanism for Biological and Machine Vision.* Stanford Press, 2002.

[9] S.R. Harker, C.R. Vogel, T. Gedeon. Analysis of constrained optimization variants of the Map Seeking Circuit Algorithm, *Journal of Mathematical Imaging and Vision*, 29 (2007), pp.49-62.

The alternative choice to a parallel implementation is a sequential procedure updating each distribution in turn, and using the updated distributions $\mathbf{p}^\ell$ and the corresponding updated superposition $P^\ell = \sum_{i=1} p_i^\ell g_i^\ell$ to compute the projected gradients of the next distribution $\mathbf{p}^{\ell+1}$. By exploiting adjoints

$$\langle \mathcal{O}, P^L \cdots P^\ell \cdots P^1 \mathcal{I} \rangle = \langle (P^\ell)^T \cdots (P^L)^T \mathcal{O}, P^{\ell-1} \cdots P^1 \mathcal{I} \rangle \qquad (11)$$

the updating procedure can be made efficient in the maximal correspondense model (15).

0.1.6 Results

Using the model developed above and projected gradients, figure 1 shows how each image point corresponds bijectively to a single point on the CAD model using this type of model.

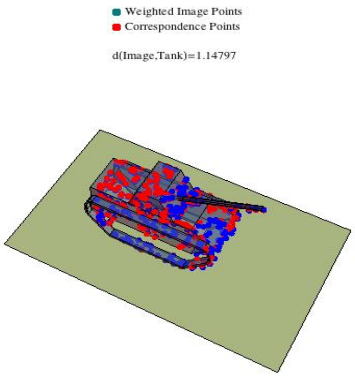


Figure 1: Correspondence between image points and correspondence points on the CAD model.

Given a whole database of objects one can sequentially apply the algorithm to each prototype (generic representative of a class of objects) and obtain results such as in figure 2 where the distances between the various objects in the database can be ordered. An alternative is to add another variable probability distribution which optimizes over all the objects simultaneously, $\{p_\kappa^{ob}\}$.

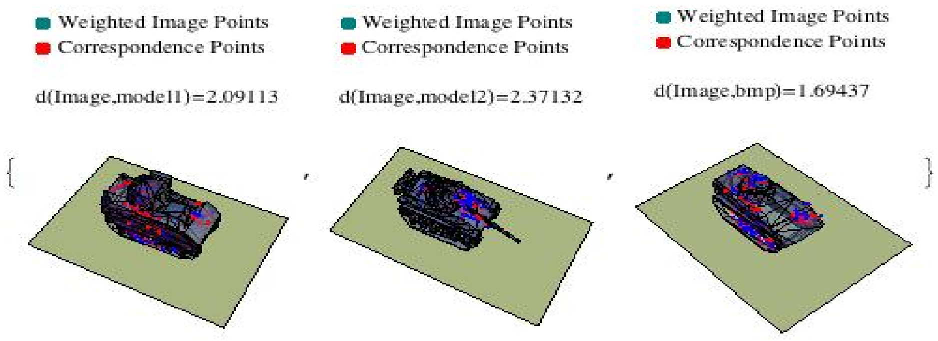


Figure 2: Metric distances between a point cloud of data and several objects.

More results are shown in the following pages where the complexity of the models is detailed in the CAD models shown. It is computationally critical to the solution algorithm that the CAD model consist of as few surfaces as possible. The existing CAD model generators are incredibly inefficient in this regard; they generate models consisting of triangles so a simple square is always represented by two surfaces rather than a single surface with 4 vertices. We have constructed a “glue model” which attempts to glue such surfaces together based upon a common edge and the surface normals being identical. Unfortunately many of the CAD model generators generate models such that the two surfaces which should have identical surface normals do not. We have attempted to correct this but it is not always possible (algorithmically) since the data can sometimes vary significantly.

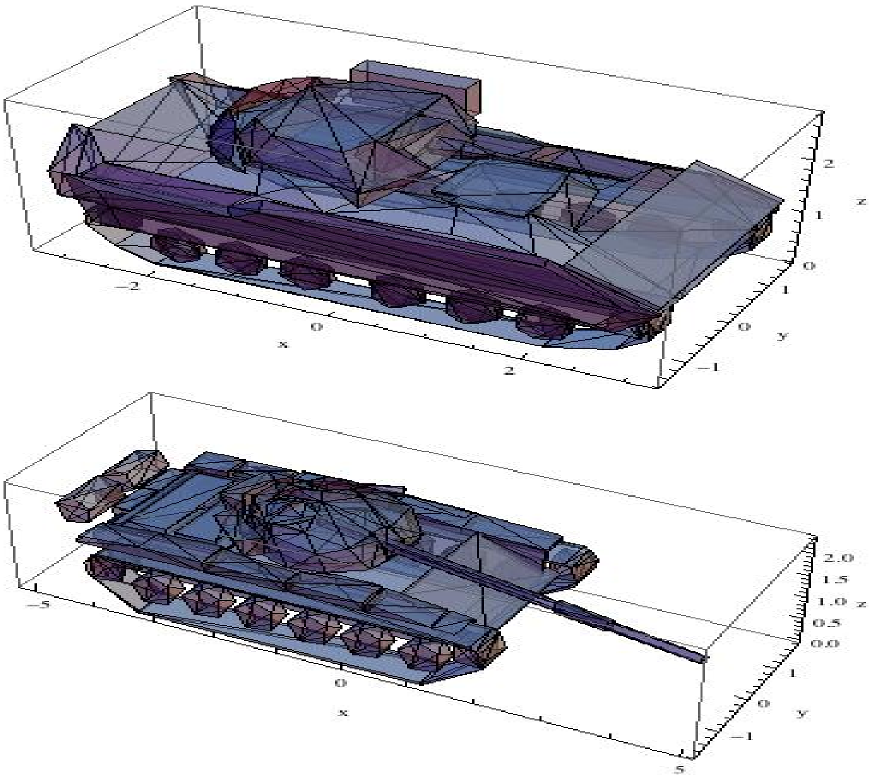


Figure 3: Some CAD models employed for testing (models 1 and 2).

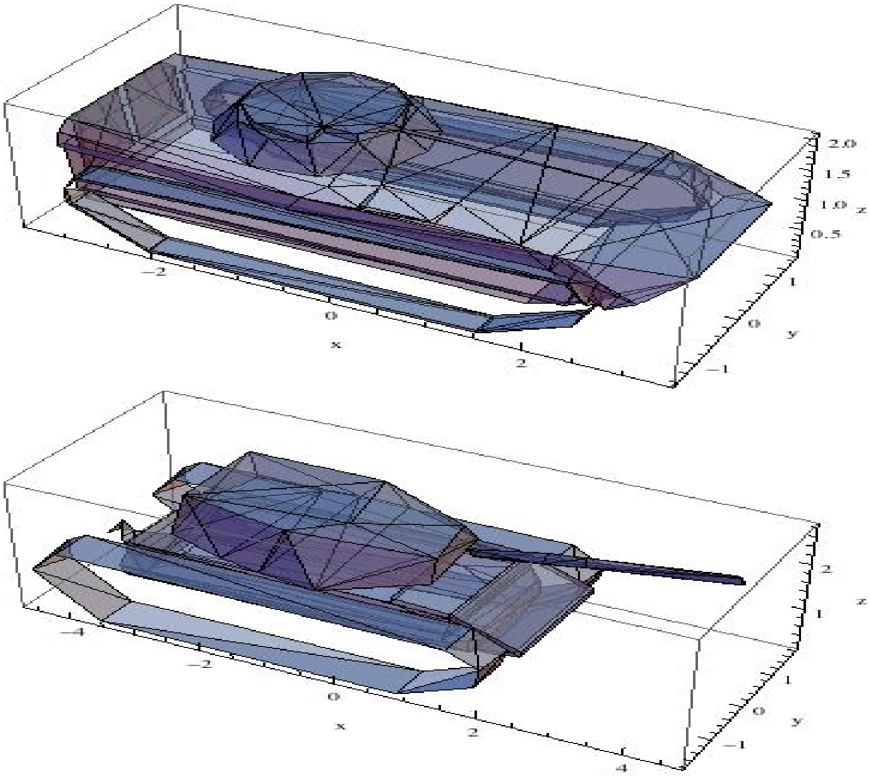


Figure 4: CAD Model 3 and 4.

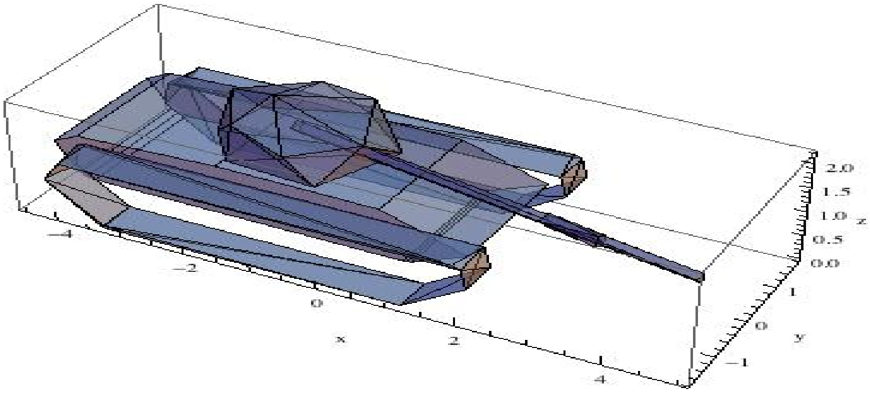


Figure 5: CAD Model 5.

The image data for Test 1, generated synthetically, is shown in Figure 6 while the results of employing this test data are shown in Figure 7. Here the image data is such that the objects can be clearly distinguished (separated). For some of the image data that follows this is not the case.

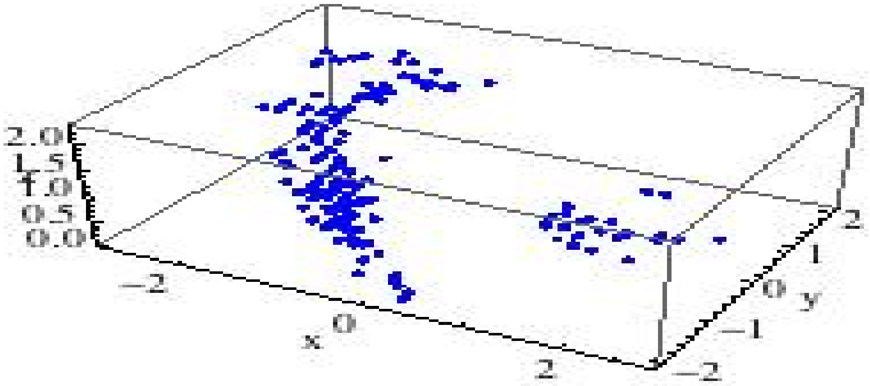


Figure 6: Image Data for Test 1.

The image data for Test 2 is shown in Figure 8 (These results are better viewed in Mathematica where the images can be rotated, scaled, etc.)

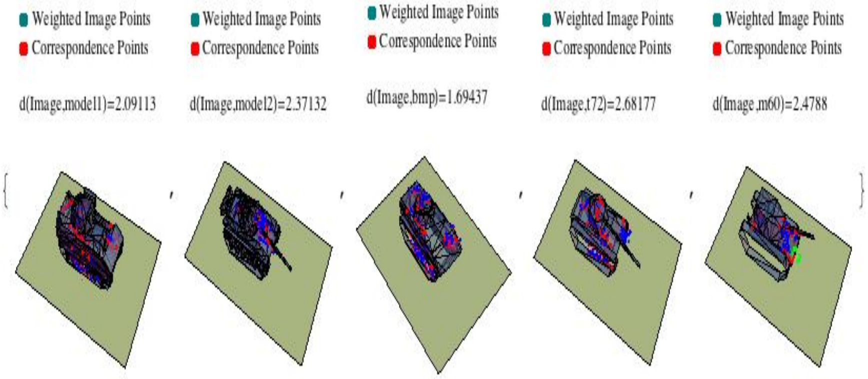


Figure 7: Results with image data set 1.

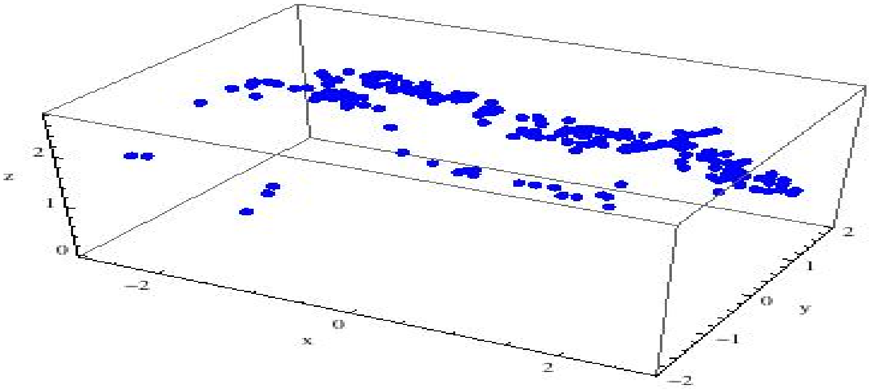


Figure 8: Image data for Test 2.

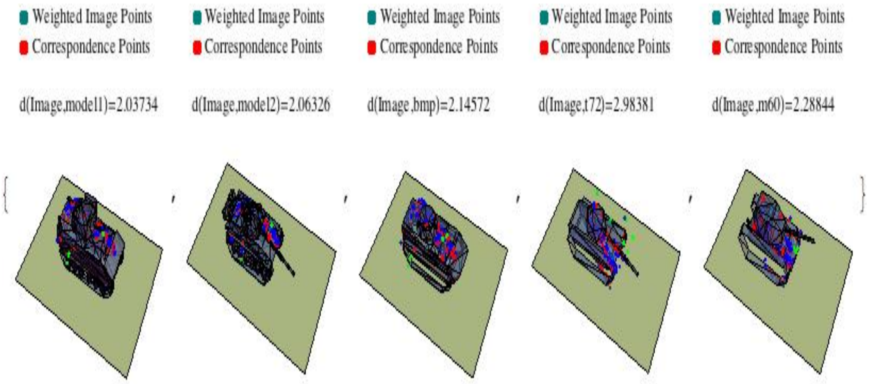


Figure 9: Results with image data set 2.

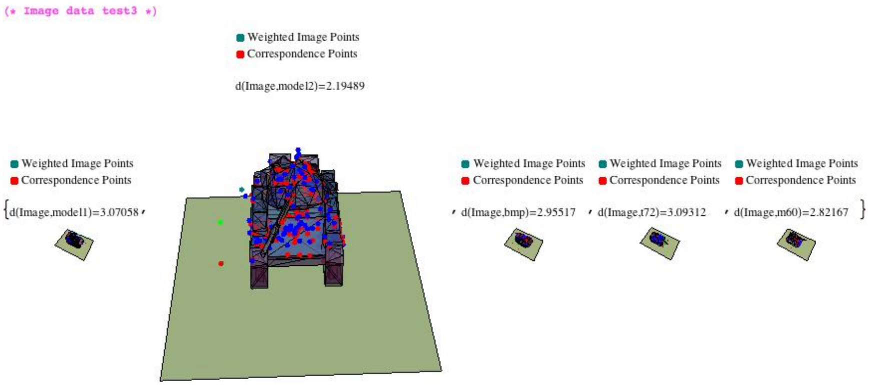


Figure 10: Results with image data set 3.

With image data set 3 there are not enough points on critical components to distinquish between the first 3 models and the results give the wrong target. We can constrast this with data set 1 where the viewpoint was more favorable to be able to separate the models and give the correct result.

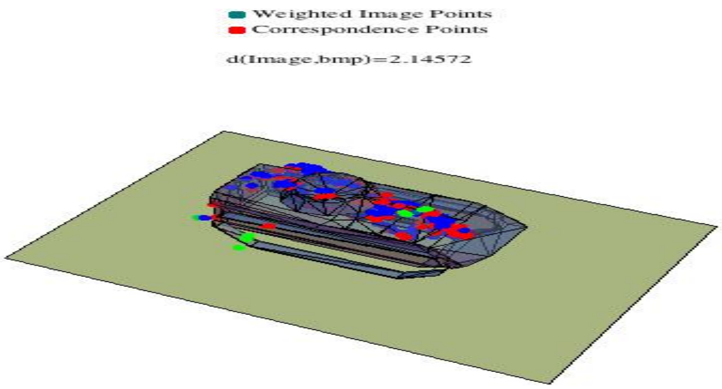


Figure 11: Results on data from bmp (correct target) with image data set 3.

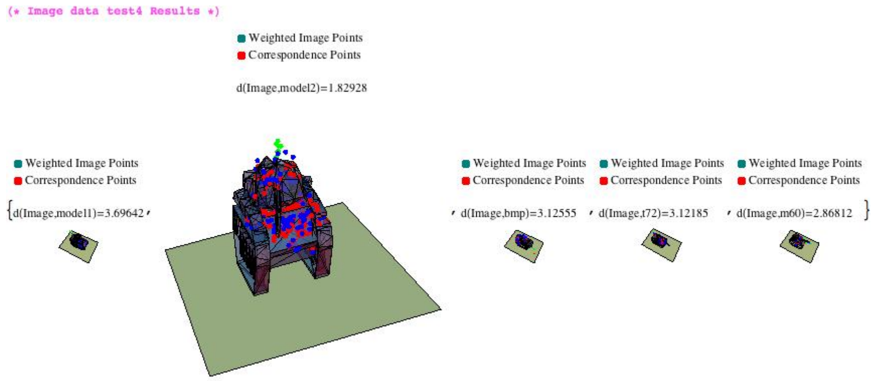


Figure 12: Results with image data set 4.

Matching points to surfaces can be modified by extracting planes of image points in the point cloud data. Then by using generalized principle component analysis, discussed in section 0.1.7, one can match the extracted surfaces to the CAD model surfaces. If one also can obtain a good estimate of where the ground plane is then the correspondence variables can be initialized accordingly and an initial "total ignorance" distribution of a uniform distribution is not necessary. Even better estimates can be obtained by taking into account the size of the various surfaces so that if surface 1 is twice as large as surface 2 then for an arbitrary image point one would expect a higher probability that the image point came from the second surface. This probabilistic interpretation (perspective) of the problem provides guidance in how to initialize all the unknown variables.

0.1.7 Generalized PCA

The technique of generalized principal component analysis allows one to extract hyperplanes from point clouds. Each one of these hyperplanes has a surface normal so that if one can estimate the ground plane then the surface normals of the extracted hyperplanes can be compared against the surface normals of the surfaces of the CAD model to come up with prior probabilities for the assignment correspondence probabilities. In Figure 13 a point cloud consisting of two clusters of points, each corresponding to a distinct hyperplane, are shown. We can associate a (normalize) surface normal, defined up to a sign, with each one of these hyperplanes. Call these surface normals s_{orange} and s_{blue}. Given a single surface with a known (normalized) surface normal $s_{objectPart}$ the correspondence between the given surface can be determined. A simple expression is given by the absolute value of the inner product between the hyperplane normals and the surfaces normals, $|\langle s_{orange}, s_{objectPart}\rangle|$ and $|\langle s_{blue}, s_{objectPart}|\rangle$ and the resulting associated probabilities based upon this are shown in Figure 14.

0.1.8 Image Point Spread Model

In the *Image Point Spread Model* the lack of knowledge of the precise location of each image point can be incorporated directly into the optimization model of the problem. This weighted least squares optimization model is

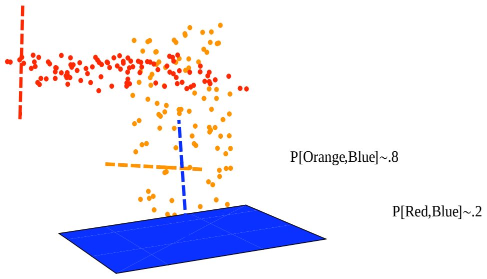


Figure 13: A point cloud consisting of two distinct hyperplanes.

$$\min_{p_{im},w_i,\alpha_{imk},R,T,T_i} \sum_{i=1}^{N_I} \sum_{m=1}^{N_S} p_{im} || R(w_i u_i) + (T + \Delta T_i) - \textstyle\sum_{k=1}^{N_V(m)} \alpha_{imk}\, x_m^k ||_F^2$$

subject to

$$|\Delta T_i| \leq \delta_i \tag{12}$$

where N_I is the number of image points, N_S is the number of "surfaces" in the CAD Model, $N_{V(m)}$ is the number of vertices of the m^{th} surface of the CAD model, and δ is a parameter expressing our uncertainty in the knowledge of the image points. While this parameter can vary for each individual image point our present implementation just assumes a constant value over all image points. Note that this model is ideal for a parallel implementation with respect to optimization over the individual point spreads ΔT_i since they are all decoupled constraints.

We have added this point spread capability to our working algorithm which coded up in Mathematica. We have applied this to our test model shown in

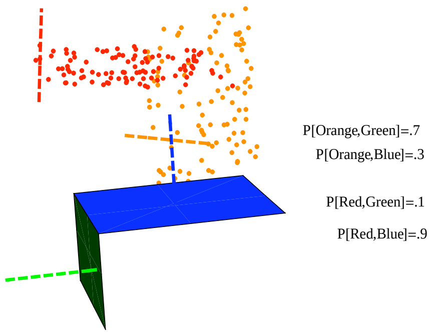


Figure 14: Computing prior probabilities based upon the surface normals of the extracted hyperplanes.

Figure 15.

Figure 16 shows the results which are exactly what we expected from this model - it decreases the weighted least squares error as the slack δ_i is increased. The x-axis in this figure represents the value δ_i and the y-axis shows the RMS error for this value.

The RMS error does not go to zero because of the discretization involved in the given model. By refining the discretization it is possible to get the error to converge to zero.

One must have relatively small errors in the signal else everything starts to look the same, i.e., any image could have been produced by any object. Here is the same image data applied to a decoy.

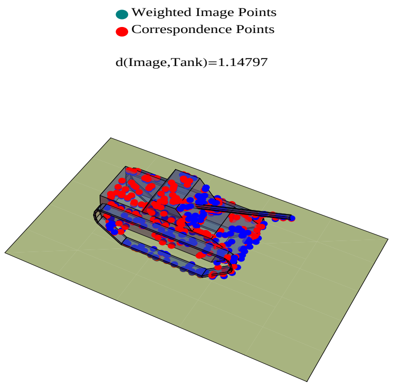


Figure 15: Standard Test Case

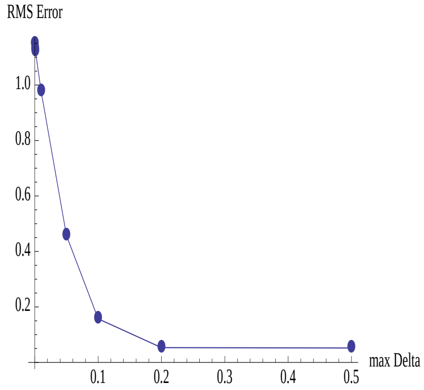


Figure 16: Image Point Spread Results

0.2 2D-3D

In the 2D-3D matching problem we only have 2 dimensional information available about the object in question. In this case the 3D-3D model can be modified so that the third unknown coordinate is treated as an unknown parameter which can be optimized over. To implement such an approach it is necessary to extract lines in the 2D data, say by a canny edge detection algorithm, and to match that up against a wire frame model of the object.

An initial manual registration of the edge detected image data to the wire frame model is employed. We start with the 2D image (so the 3rd coordinate z

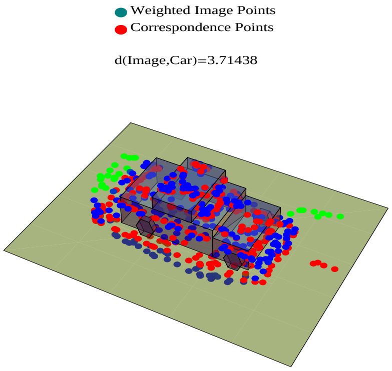


Figure 17: Car Decoy Matching Correspondence - $\delta = 0$

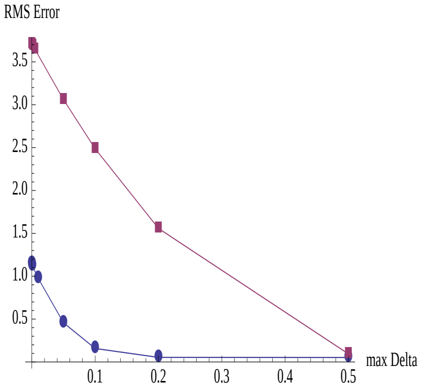
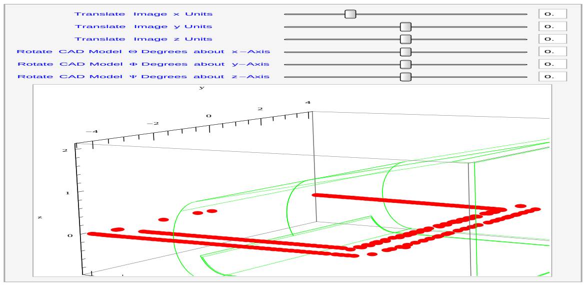


Figure 18: Image Point Spread Results for Tank and the Car Decoy

is some constant for all points). Figure 19 shows the arbitrary initial placement.

Figure 20 shows a manual registration obtained by translating the image and rotating the wire frame model.

This initial registration need not be "exact" just close so as to avoid issues of global optimality concern. By this initial registration the code can handle



Translate Image x Units 0.

Translate Image y Units 0.

Translate Image z Units 0.

Rotate CAD Model Θ Degrees about x−Axis 0.

Rotate CAD Model Φ Degrees about y−Axis 0.

Rotate CAD Model Ψ Degrees about z−Axis 0.

Figure 19: Here is the image and wire frame object - need an initial manual registration.

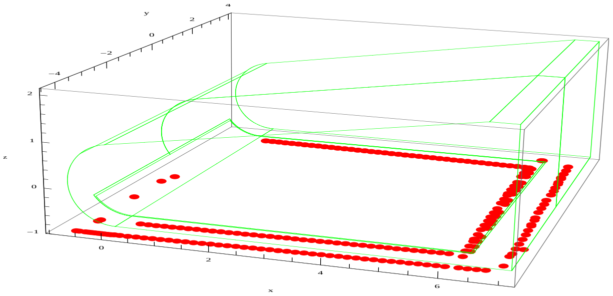


Figure 20: An initial alignment.

relatively large CAD models because the initial probabilities can be set accordingly.

The optimization model for this problem is

$$
\begin{array}{ll}
\min\limits_{R\in SO(3), I, T\in\Re^2, p_{i,j}, \alpha_{i,j,k}} & \sum_i \sum_j p_{i,j}||(Iu_i + T) - \pi \sum_k R(\alpha_{i,j,k} x_j^k)||^2 \\
\text{subject to} & \\
\quad \sum_j p_{i,j} = 1 \quad \forall i & \\
\quad p_{i,j} \geq 0 \quad \forall i,j & \\
\quad \sum_k \alpha_{i,j,k} = 1 \quad \forall i,j & \\
\quad \alpha_{i,j,k} \geq 0 \quad \forall i,j,k &
\end{array}
\tag{13}
$$

where π is the projection mapping onto 2D (wolog, say the first two coordinates). Now the object, rather than the image, is rotated and the transformation of the image involves an inversion I to account for parallax - this of course depends upon the model - which occurs with most cameras.

This initial registration gives me a good starting point for the optimization problem - that being to find the optimal general linear projective[10] transformation of the image data and optimal 3D rotation of the wire frame model to minimize the total error. The result is shown in Figure 3 which gives the image data (blue pts) and corresponding points (red points) on the wire frame model -

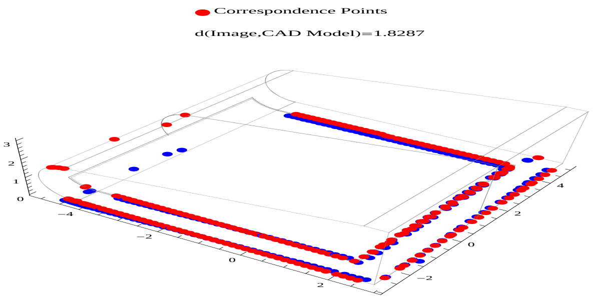


Figure 21: Here is after the optimization routine improves the initial registration (alignment).

By using finer discretizations I can obtain more accurate results - as expected. This refinement problem will need some analysis.

This result allows me to then properly place the damaged image data onto the 3D CAD model

The optimization routine then calculates the corresponding points on the 3D CAD model. Here's the optimal correspondence

[10] The scale is limited to a small range about 1 so the metric still exist (is nonzero).

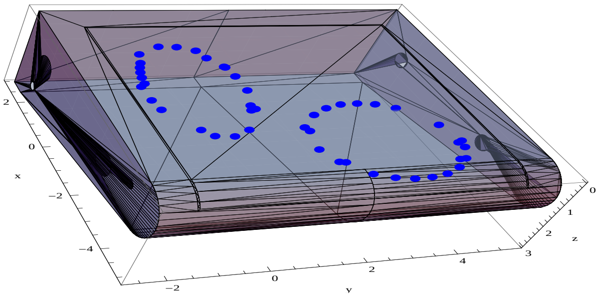


Figure 22: Here is the registration of the damaged data points onto the 3D CAD Model.

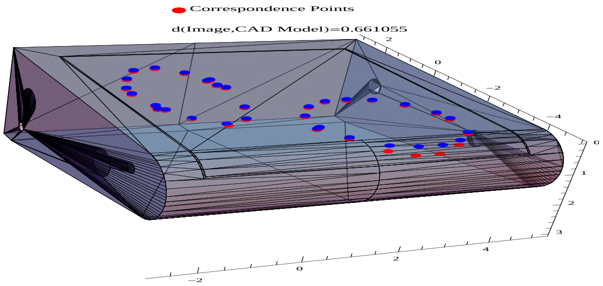


Figure 23: Here is the correspondence between the damaged data points and correspondence points on the 3D CAD Model.

These red correspondence points “are the solution”. Note that the blue image points don’t lie exactly on the surface - this is because the ”z” coordinate is unknown. A good initial alignment is desirable to get the image points to lie as close as possible to the surface but is not required. Indeed for a curved surface some of the image points will necessarily lie off of the surface. You can see this in the above image on the front curved surface. The picture below illustrates this.

’

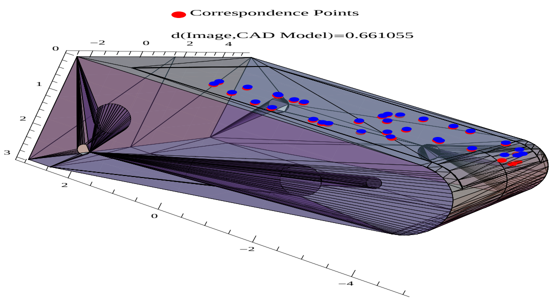


Figure 24: Another view of the correspondence between the damaged data points and correspondence points on the 3D CAD Model.

0.3 Rubiks Cube

Rubik's Cube can be represented by specifying a state of the cube by specifying the location of 20 cubes and the orientation of those cubes. This can be accomplished with a $(0,1)$-matrix representation. For Rubik's Cube the transformation group is the set of all possible permutations of the cube which is a discrete group. This transformation group $\mathcal{G}$ itself can be decomposed into 'layers' and the rotations about each of the six faces, $g = P^l_{f,t} \circ \ldots \circ P^2_{f,t} \circ P^1_{f,t}$.

Let $\mathcal{G}^l$ be the group of permutations on the cube, for each layer l of rotations, yielding the model

$$\begin{array}{ll} \min\limits_{p^l_{f,t}} & \frac{1}{2}\prod_{l=1}^{L}\sum_{f=1}^{6}\sum_{t=0}^{3} p^l_{f,t}||\mathcal{O} - P^L_{f,t}P^{L-1}_{f,t}\cdots P^1_{f,t}\,\mathcal{I}||^2 \\ \text{subject to} & \\ \sum_{f=1}^{6}\sum_{t=0}^{3} p^l_{f,t} = 1 & \text{for all layers } l \\ p^l_{f,t} \geq 0 & \text{for all faces } f\text{, layers } l\text{, and turns } t \end{array} \tag{14}$$

where L is the number of layers of rotations. Upon further simplification the objective function can be written as $\min\limits_{p^l_{f,t}} \frac{1}{2}\prod_{l=1}^{L}\sum_{f=1}^{6}\sum_{t=0}^{3} p^l_{f,t}||\mathcal{O} - \prod_{m=1}^{L} P^m_{f,t}\,\mathcal{I}||^2$.[11]

[11] For computational purposes this is not the most efficient model since for $t = 0$ turns about each face, $P^l_{f,t}$ is the identity map. A single identity map suffices for each layer yielding 19

Using the fact $||x||_2^2 = \langle x, x\rangle$ and the normalization constraint on the set of variables $\{p^l_{f,t}\}_{f,t}$, upon expansion of the squared objective function and dropping constant terms in the expression to be minimized we obtain the maximal correspondence model with objective function

$$\max_{p^l_{f,t}} \quad \prod_{l=1}^{L} \sum_{f=1}^{6} \sum_{t=0}^{3} p^l_{f,t} \langle \mathcal{O}, \prod_{m=1}^{L} P^m_{f,t} \mathcal{I} \rangle \tag{15}$$

subject to the same linear constraints as above. Hence, as stated above for the general least squares formulation, the minimuml error and maximal correspondence formulations are equivalent Our algorithmic results for Rubiks Cube are identical to those presented in *Analysis of Constrained Variants of the Map-Seeking Algorithm*[12]. For comparative purposes it should be noted that in that paper they define a 'layer' to consist of at most a single 90 degree counter clock wise (ccw) rotation about any face (yielding 7 possible mappings at each layer) whereas we define a layer to be any rotation about any of the faces (19 possible mappings) and the term "multilinear model" will refer to either form. This multilinear model should not to be implemented directly as it is written. The bilinearity of the inner product allows the summation terms to be brought inside the brackets and permits the formation of the "superposition of transformations" at layer l by $P^l = \sum_{f,t} p^l_{f,t} P^l_{f,t}$ to yield the formulation with objective function

$$\max_{p^l_{f,t}} \quad \langle \mathcal{O}, \prod_{m=1}^{L} P^m \mathcal{I} \rangle \tag{16}$$

which greatly reduces the computational effort required to solve the problem. The formulation of recognition problems directly using this model is what the Map-Seeking Algorithm is based upon. By using a weighted listing, with real coefficients restricted to $[0, 1]$, of all possible combinations embeds the problem

unique possible mappings for each layer considered rather than the $24 = 6 \times 4$ possibilities in the given model. We have chosen the above model as it provides an elegant and simple formulation for exposition.

[12] Harker,S., Vogel C., Gedeon T., Journal of Mathematical Imaging and Vision. Volume 29 Issue 1, pp 49-62.

into the real domain for computational advantage and allows for the probabilistic interpretation[13] which allows for further exploitation[14]. At a foundational level this model is the multilinear least squares error formulation (14). In the 3D recognition problem the superposition of transformations involves both superpositions of rotations and superpositions of translations. The exploitation of the bilinearity changes the complexity of the problem from multiplicative to additive.

0.3.1 The MultiLinear Model Optimization Updating Step

The *decoupled linear constraints* associated with the model allows one to update the probabilities associated with each layer in parallel which is considerably more efficient than a sequential procedure. Using a projected gradient technique to calculate a search direction d^l at each layer the 19 probabilities associated with the 19 distinct rotational mappings of the cube at each layer can be simultaneously updated via a step $p^l = p^l + \alpha d^l$. One can either use a line search procedure to calculate the step length α or, upon normalization of the search direction d, use a constant step length α at each iteration. Our experience suggest the constant step length is computationally just as efficient as a line search procedure.

The alternative choice to a parallel implementation is a sequential procedure updating each layer in turn, and using the updated probabilities at layer l, P^l in the calculation of the projected gradients at layer $l+1$. By exploiting adjoints

$$\langle \mathcal{O}, P^L \cdots P^l \cdots P^1 \mathcal{I} \rangle = \langle (P^l)^T \cdots (P^L)^T \mathcal{O}, P^{l-1} \cdots P^1 \mathcal{I} \rangle \tag{17}$$

the updating procedure at layer l can be made efficient.

Here is a sequence of distributions showing the convergence to dirac measures for a 6-layer problem. In the software this information is animated. The first image of the sequence shows a uniform distribution for all 6 layers. As the

[13] Note that a solution can always be found at a dirac probability measure.

[14] These additional exploitations based upon the probabilistic interpretation will be discussed in a future paper

optimization algorithm proceeds all the distributions approach a dirac measure fairly uniformly since they are all being updated simultaneously. (That code was run in parallel.)

0.3.2 Rubiks Cube Results

The parallel and sequential implementation using projected gradients for the search direction produced the same success rate in the determination of the global minimum. This is worthy of note since the two procedures actually approach the solution along a potentially different trajectory.

These results are shown in the bottom graph of Figure 1 as a function of a varying number of layers. For an n layer problem we generated a random initial mixup of the cube consisting of n random rotations of the cube. Each random rotation was one of the 3 distinct nontrivial rotations about a given (random) face.

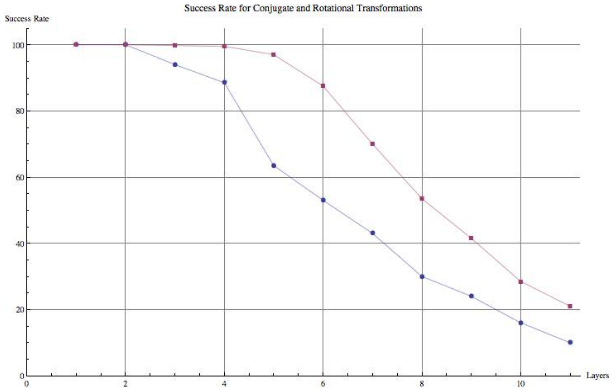


Figure 25: Success rate of the algorithm for the two methods involving search directions of simple permutations and the other using conjugates for the search directions.

0.3.3 Conjugates and Local minimum

In analyzing those problems where the optimization algorithm failed to find the global minimum we found a striking number of those cases occurred when the initial mixup of the cube involved a conjugation. This phenomena is illustrated in Figures 29 which shows the green face rotated 3 times counterclockwise (270 degrees) followed by a rotation of the white face by two turns, and finally a rotation of the green face by 1 counterclockwise rotation.

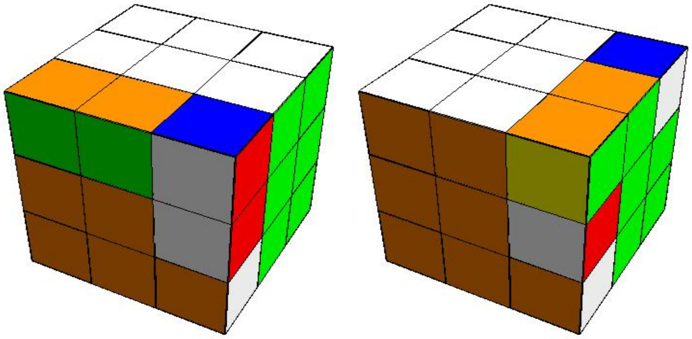


Figure 26: The cube on the left shows an initial conjugation of the Green and White faces. The cube on the right shows the local minimum obtained by using elemenatary rotations.

Such a rotation can be denoted by ghg^{-1} with g being the group element "rotate the green face 3 times". The inverse of this element, g^{-1}, is "rotate the green face once" since four rotations about any face gives back the identity map (no rotation at all). Associated with these conjugations are local minimum which have a very large basin of attraction so unless one starts very close to the global optima one gets trapped in a local minina using projected gradients as a search direction. This phenomena extends to any size problem. If a conjugation occurs anywhere within the initial mixup of the cube, say the transformation applied to the cube is $g_1 h_1 h_2 h_3 g_1^{-1} (= ghg^{-1})$ then a local minimum will exist with a large basin of attraction. The noncommutativity of the transformation group causes difficulty as using projected gradient will simply yield the solution of h^{-1} rather than $gh^{-1}g^{-1}$. To the optimization algorithm the composite transformation $ghg^{-1} \approx gg^{-1}h = h$. An analogy in 1D would be a function as shown in Figure 3 where a gradient based algorithm will inevitably tend to the local minimum.

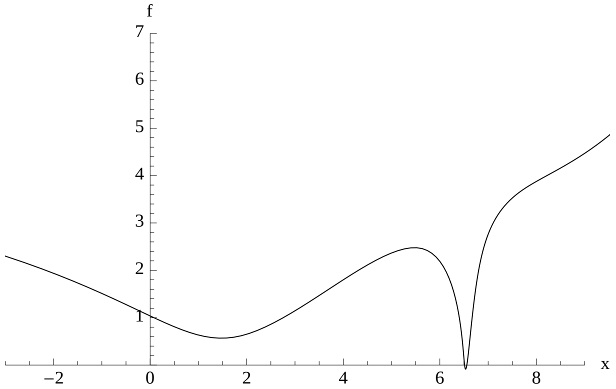


Figure 27: A function with a local minimum having a large basin of attraction.

One can improve the performance of the algorithm by directly employing conjugations in the model itself. In the objective function (29) the terms P^l are superpositions of rotations about the six faces of the cube. This objective function can be altered by letting each superposition P^l be a superposition of conjugations. This change in the model simply says one can obtain a solution using conjugations, which includes all the simple rotations since for g and h being rotations on opposite face $ghg^{-1} = h$, and is such that a projected gradients algorithm applied to it can avoid more local minimum than the simple model.

Because the number of possible permutations is increased from 19 distinct possibilities to 235 distinct possibilities one may expect the solution to be determined by searching over a larger space to yield a better result. Using conjugations as search directions this is true. However there are many other choices, such as choosing commutators $ghg^{-1}h^{-1}$ which when used in conjunction with conjugations yield 485 distinct permutations, and using this larger set yields poorer results than when using only conjugations. Simply enlarging the number of possible permutations does not guarantee the algorithm will find a better solution[15]

0.3.4 Conjugate Search Results

Using the above multilinear model generating random initial images, based on 200 random samples and an initial uniform distribution, we obtained the following success rates for varying number of layers as displayed in the top graph of Figure 1. In Figure 4 the computation times show how the computation times increase roughly linearly with respect to the number of layers. The computational times are included to show relative computation times - the code has not optimized for speed.

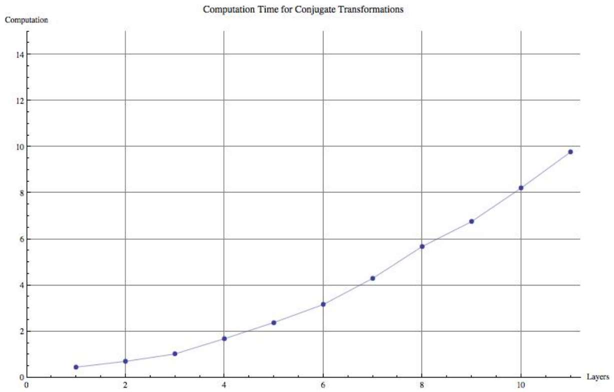


Figure 28: A function with a local minimum having a large basin of attraction.

[15] In the extreme case one can take the superposition to be a linear combination of every discrete possibility so that only one layer is required. In this case the problem reduces to a linear programming problem which is convex so the global optima can always be determined. However this is not computationally feasible since the number of such permutations is exceedingly large.

0.4 DC Programming Model

One of the challenges in attempting to solve any object recognition problem using optimization resides in the difficulty of determining a global optimum to the model one employs for modeling the problem. If the algorithm used to solve the optimization model determines a local minimum such a formulation is useless. Since there is no optimization algorithm which guarantees a global minimum for all models it is desirable to choose the optimization model such that it is a convex model (a convex objective function with convex constraints) whereby it is possible to guarantee a global minimum since for a convex optimization model any local minimum is unique and hence a global minimum[16].

In recent years there has been an increasing awareness that convex and quasi-convex optimization models can play an important rule in recognition problems. To our knowledge the general recognition problem, which does not assume a known correspondence or knowledge of the transformation mapping, has not been addressed with convexity (global optimality issues) in mind. The Map Seeking Circuit (MSC) algorithm of Arathorn[17] has achieved some success in addressing the global optimality problem and uses the concept of superposition to obtain a computationally feasible model for recognition problems. This concept follows directly from the discretization of the problem which gives a multilinear programming problem and the bilinearity of the inner product used for the objective function. Our modeling procedure does not require discretization of space, addressing the correspondence problem as an assignment problem, and uses a minimum error model followed by a transformation of variables. We show the relationship between the maximal correspondence model[18] employed by MSC and the minimum error model. In our models the unknown variables can be interpreted as probabilities which allows for additional techniques to be

[16] The uniqueness is in terms of the objective function value; the argument of the minimum is not necessarily unique.

[17] Map Seeking Circuits in Visual Cognition: A Computational Mechanism for Biological and Machine Vision. Stanford Press, 2002.

[18] The maximal correspondence formulation is a multilinear programming problem which is non convex.

brought to bear on the problem.

When a convex optimization model is not available a good alternative is to construct the model using a difference of convex (DC) functions.

Towards this end we formulate shape based object recognition problems using an optimization model with a DC objective function and convex constraints. Such a procedure provides a theoretical foundation for further research.

This paper is organized into two sections. In Section 1 basic modeling of shape based recognition problems is addressed using discretization of the Euclidean transformation variables. This modeling yields a multilinear programming problem which can be transformed into a DC optimization problem using the exponential transformation of variables.

Section 2 consist of applications including Rubik's Cube, point-to-point matching problems, and the generalization to point to surface matching. These applications illustrate the implementation of the modeling procedure and highlight computational considerations and aspects. The modeling of the feature correspondence problem as an assignment problem[19] is implemented and explained on a point to point matching problem.

0.4.1 Discretizing a Model

Any continuous optimization problem can be discretized over the variables to obtain a "probabilistic model".

Consider the two variable problem

$$\min_{x,y} f(x,y) \tag{18}$$

where f is any continuous function in both arguments.

Discretizing the variable x into N_x possible values $\{x_1, x_2, \ldots, x_{N_x}\}$ and the variable y into N_y values $\{y_1, y_2, \ldots, y_{N_y}\}$ problem (18) can be approximated as

[19] The assignment problem is well known in the Linear Programming community and can be found in most book on linear programming.

$$\begin{aligned}
&\min_{p_{x_i},p_{y_j}} \sum_{i=1}^{N_x}\sum_{j=1}^{N_y} p_{x_i} p_{y_j} f(x_i, y_j)\\
&\text{subject to}\\
&\sum_{i=1}^{N_x} p_{x_i} = 1\\
&\sum_{j=1}^{N_y} p_{y_j} = 1\\
&p_{x_i} \geq 0 \quad \text{for } i = 1, \ldots, N_x\\
&p_{y_j} \geq 0 \quad \text{for } j = 1, \ldots, N_y
\end{aligned} \tag{19}$$

where the vectors $P_x = \{p_{x_1}, p_{x_2}, \ldots, p_{x_{N_x}}\}$ and $P_y = \{p_{y_1}, p_{y_2}, \ldots, p_{y_{N_y}}\}$ can be *interpretted* as probability distributions. The global solution to problem (19) will occur at dirac probability measures $P_x = \delta_{\hat{i}}$ and

$P_y = \delta_{\hat{j}}$ where f achieves a minimuml value at the values $x_{\hat{i}}$ and $y_{\hat{j}}$ among all the discretized values. This formulation is simply a crude method for *looking at all possible (discretized) values* and choosing the minimum value. This idea can be applied to any unconstrained or decoupled constrained optimization problem. It turns an $n-variable$ problem into an $n-linear$ programming problem which is linear in each probability $P_\cdot$. The formulation (19) is not computationally feasible for a problem with a large number of probabilities and/or discretizations unless the objective function f has the property of bilinearity possessed by inner products. In such a case the summation terms can be brought inside the inner products thereby reducing the computational complexity from multiplicative in the number of variables, $N_x N_y$, to additive, $N_x + N_y$.

In modeling rigid body motion the minimum norm error squared model defined by the Euclidean-invariant function

$$d(Ob, Im) = \min_{R,T} \quad ||Ob - (R(Im) + T)||^2 \tag{20}$$

where R is a an orthogonal rotation and T is a translation, is an appropriate model and determines a pseudo metric distance between an object and an image. Discretizing this problem (2D for illustrative purposes)

$$
\begin{aligned}
d(Ob, Im) &= \min_{p_{R_j}, p_{x_k}, p_{y_l}} \sum_{j=1}^{N_R}\sum_{k=1}^{N_x}\sum_{l=1}^{N_y} p_{R_j} p_{x_k} p_{y_l} ||Ob - (R_j(Im) + T_{k,l})||^2 \\
&= \min_{p_{R_j}, p_{x_k}, p_{y_l}} \langle Ob, Ob\rangle - 2\langle Ob, \textstyle\sum_{j=1}^{N_R} p_{R_j} R_j(Im) + \sum_{k=1}^{N_x}\sum_{l=1}^{N_y} p_{x_k} p_{y_l} T_{k,l}\rangle + \\
&\quad 2\langle \textstyle\sum_{j=1}^{N_R} p_{R_j} R_j(Im), \sum_{k=1}^{N_x}\sum_{l=1}^{N_y} p_{x_k} p_{y_l} T_{k,l}\rangle + \sum_{k=1}^{N_x}\sum_{l=1}^{N_y} p_{x_k} p_{y_l} \langle T_{k,l}, T_{k,l}\rangle + \langle Im, Im\rangle
\end{aligned}
\tag{21}
$$

where R_j, x_k, and y_l are the discretized rotation and translation values. This can be further simplified since $T_{l,k} = \{x_k, y_l\}$.

0.5 The DC Programming Formulation

The multilinear programming model formulation of the problem as illustrated in (19) is still inherently difficult to solve. The higher the multilinearity the more difficult the problem is to solve using traditional algorithms. However this multilinear programming problem, which is nonconvex, can be transformed into a difference of convex functions as follows:

Note that the equality constraint in the multilinear model, $\sum_{i=1}^{N_x} p_{x_i} = 1$, is equivalent to $||P_x||_1 = 1$ since each $p_{x_i} \geq 0$. (Any l_p norm, $p \geq 1$ can be used in the model.) This is a normalization condition which is neccessary to avoid the trivial solution $P_x = P_y = \mathbf{0}$.

This condition on the variables $p.$ can be achieved by scaling each such variable by its' norm so the multilinear model above is equivalent to

$$
\begin{aligned}
&\min_{p_{x_i}, p_{y_j}} \sum_{i=1}^{N_x}\sum_{j=1}^{N_y} \tfrac{p_{x_i}}{||P_x||_1} \tfrac{p_{y_j}}{||P_y||_1} f(x_i, y_j) \\
&\text{subject to} \\
&p_{x_i} \geq 0 \quad \text{for } i = 1, \ldots, N_x \\
&p_{y_j} \geq 0 \quad \text{for } j = 1, \ldots, N_y
\end{aligned}
\tag{22}
$$

which eliminates the equality constraints. Factoring out the norms since they are independent of the summation indices and taking the natural logarithm, ln, of the objective function gives

$$\begin{aligned}
&\min_{p_{x_i},p_{y_j}} ln(\sum_{i=1}^{N_x}\sum_{j=1}^{N_y} p_{x_i}p_{y_j}f(x_i,y_j)) - ln(||P_x||_1) - ln(||P_y||_1) \\
&\text{subject to} \qquad (23)\\
&p_{x_i} \geq 0 \quad \text{for } i = 1,\ldots,N_x \\
&p_{y_j} \geq 0 \quad \text{for } j = 1,\ldots,N_y
\end{aligned}$$

For each probability component variable p_i define

$$q_i = ln(p_i) \qquad (24)$$

where $ln[0] \equiv -\infty$. Thus $e^{q_i} = p_i$ which transforms the multilinear programming problem (22) to

the equivalent problem

$$\begin{aligned}
&\min_{q_{x_i} q_{y_j}} ln(\sum_{i=1}^{N_x}\sum_{j=1}^{N_y} e^{q_{x_i}+q_{y_j}} f(x_i,y_j)) - ln(\sum_{i=1}^{N_x} e^{q_{x_i}}) - ln(\sum_{j=1}^{N_y} e^{q_{y_j}}) \\
&\text{subject to} \qquad (25)\\
&q_{x_i} \leq 0 \quad \text{for } i = 1,\ldots,N_x \\
&q_{y_j} \leq 0 \quad \text{for } j = 1,\ldots,N_y
\end{aligned}$$

Assuming the function $f(x_i, y_j) \geq 0$, for all indices i and j, the objective function is a difference of convex functions. This critical property is satisfied in our object recognition applications because the objective function is an invariant metric defined by a norm $|| \cdot ||^2 \geq 0$.

In this model the objective function is a difference of two convex function, since $\sum_{i=1}^{N_x} e^{q_i^x}$ is a convex function and the ln operator preserves convexity, and the nonpositive constraints $q_{x_i} \leq 0$ are convex.

An optimization problem where the objective function and constraints are a difference of convex functions is referred to as a DC (difference of convex functions) programming problem.

The strategy in DC programming algorithms for such a problem is to solve a sequence of convex optimization problems where for each problem one replaces the nonconvex terms, in this case the terms $ln[\sum_{i=1}^{N_x} e^{q_{x_i}}]$ and $ln[\sum_{i=j}^{N_y} e^{q_{y_j}}]$, by their affine approximation which is the first two terms in the Taylor series expansion of the function evaluated at the previous iterate. The constant term in the expansion can be dropped since any constant term in the objective function does not change the argument of the minimum. Thus for a general DC objective function

$$z(x) = f(x) - g(x) \tag{26}$$

one uses the objective functions

$$z(x^k) = f(x^k) - \nabla_x(g)|_{x^{k-1}} x^k \tag{27}$$

where x^{k-1} is an optimal solution to the previous problem in the sequence. For the first iterate $k = 1$ any feasible point suffices for x^0. This is the strategy in the primal-dual subgradient method proposed by Tao and An[20]. The dual variables of the q_i variables in this method for the above problem are precisely the probability components, $\frac{e^{q_i}}{\sum_{i=1}^{N} e^{q_i}}$. Such an algorithm applied to shape based recognition problems would use pruning (eliminating variables) along the way for computational efficiency.

1 Theory

1.1 Moving Beyond Deterministic Mappings

In the previous technical report we discussed the "Least Squares Recognition Problem" and its' application to the Rubiks cube problem. The "Least Squares Recognition Problem" essentially is the model involving the *Frobenious* norm minimization of the error between an Object $\mathcal{O}$ and an image $\mathcal{I}$, and a transformation group $\mathcal{G}$,

[20] Pham Dinh Tao, Le Thi Hoai An. D.C. Optimization Algorithms for Solving the Trust Region Subproblem. SIAM Journal of Optimization, V.18, (1998) pp 476-505.

$$\min_{g \in \mathcal{G}} ||\mathcal{O} - g\mathcal{I}||_F^2 \tag{28}$$

Essentially all our recognition problems can be abstractly viewed via this model. The process of discretization allows the problem to be stated equivalently as

$$\max_{p^l_{f,t}} \quad \langle \mathcal{O}, \prod_{m=1}^{L} P^m \, \mathcal{I} \rangle \tag{29}$$

where for layer l, $P^l = \sum_{f,t} p^l_{f,t} P^l_{f,t}$ is a superposition a components. (See the previous report for the details here, i.e., f=”face”, t=”turns”, etc.)

This model led to the results

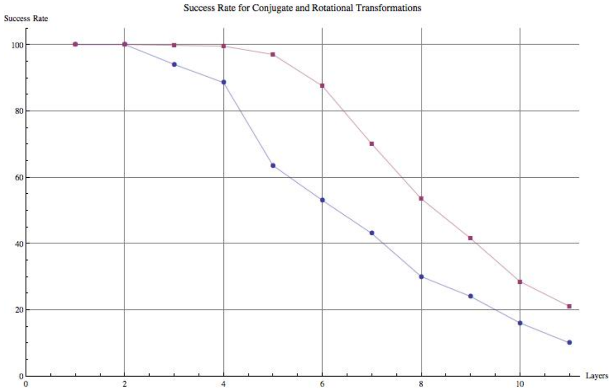


Figure 29: Success rate of the algorithm for the two methods involving search directions of simple permutations and the other using conjugates for the search directions.

which are less than spectacular in terms of finding the global minimum. While using conjugates of rotations in the superpositions helps, for large layers it too tends to fail in finding the global minimum and consequently a new idea is required. We have tried using penalty methods whereby if we obtain a nonglobal solution we simply add a penalty term to the objective function which forces the algorithm to avoid converging to that same point when the algorithm is ran again with the new objective function. While this “works” it tends to require an excessive number attempts - it has to repeatedly add new components to the penalty component. In short it ends up searching a large portion of the search space.

There is a potentially better way to address this nonglobality issue by exploiting an aspect of the probabilistic interpretation which the discretization introduces. Namely there is an underlying theoretical aspect which we discovered previously which actually led us to this new least squares formulation and the recognition that the problem is in fact equivalent to the Maximal Correspondence Problem. That theorem is the categorical result of

$$\mathcal{P}_{det} \dashv \mathcal{P}_{prob} \tag{30}$$

where $\mathcal{P}_{det}$ is the category of deterministic mappings (essentially what we are employing right now) and $\mathcal{P}_{prob}$ is the category of probabilistic mappings. Our current model and algorithmic solution does not use this fact outside the observation that we can embedd the (discrete) combinatorial problem into the continuous domain of (the category of) Probabilistic Mappings. It is this aspect which we are now trying to further exploit now that we have an understanding how it translates into actual computations.

What we have determined is that to work in the category of probabilistic mappings is that in the key concept revolves around the superposition $P^l = \sum\limits_{f,t} p^l_{f,t} P^l_{f,t}$. The maps (matrices) $P^l_{f,t}$ are all permutation maps which are *deterministic*. To work in the category of probabilistic mappings we should be employing stochastic matrices which are more general than simple permutations. A stochastic matrix is a square P such that each of its’ rows sums to one. (Permutations matrices are a special instance.) The problem with permutation matrices appears to be that they require the convergence to take place along certain search directions which “pass over hills” before reaching the valley (global minimum). By enlarging the search space (number of search directions) to include stochastic matrices convergence to a global minimum should be able to take a “circuitous route” to the global minimum. In the Theory of Markov Processes the stochastic matrices are viewed as transition matrices - they allow you to transition from one probability to another without moving along a deterministic route. In practice this approach fails because the basis set is extremely large and the procedure requires excessive computational time to be able to effectively search that space. It is a stochastic method which like all such methods requires a very large number of attempts to find the global minimum with no quarantee that the result one gets is indeed the global minimum.

1.2 The Category of Probabilistic Mappings $\mathcal{P}$

In the category of probabilistic mappings the objects are measurable spaces $(X, \mathcal{B})$, where X is a set and $\mathcal{B}$ is a σ-algebra on X. A morphism between two such objects

$$(X, \mathcal{B}) \xrightarrow{T} (Y, \mathcal{B}') \tag{31}$$

is a bivariate function

$$T : X \times \mathcal{B}' \to [0, 1] \tag{32}$$

such that for each fixed $x \in X$, the induced map defined by

$$\begin{array}{rcccl} T_x & : & \mathcal{B}' & \to & [0,1] \\ & : & B' & \mapsto & T(x, B') \end{array} \tag{33}$$

is a probability measure on the measurable space $(Y, \mathcal{B}')$, and such that for each fixed $B' \in \mathcal{B}'$, the induced map defined by

$$\begin{array}{rcccl} T_{B'} & : & X & \to & [0,1] \\ & : & x & \mapsto & T(x, B') \end{array} \tag{34}$$

is a measurable function on the measurable space $(X, \mathcal{B})$. The composite of two probabilistic mappings

$$(X, \mathcal{B}) \xrightarrow{T} (Y, \mathcal{B}') \xrightarrow{U} (Z, \mathcal{B}'') \tag{35}$$

is defined by

$$[U \circ T](x, B'') = \int_{y \in Y} U_{B''}(y) dT_x \tag{36}$$

Observe that $U_{B''}$ is a real valued function on Y,

$$U_{B''} : Y \to [0,1].$$

The *m-expectation* of f, for f any real valued function f and m any measure, is defined by

$$E_m(f) \triangleq \int f\, dm. \tag{37}$$

Consequently the composite $[U \circ T](x, B'')$ is the T_x-expectation of $U_{B''}$,

$$[U \circ T](x, B'') = E_{T_x}(U_{B''}). \tag{38}$$

This law of composition is associative.

Every measurable mapping

$$(X, \mathcal{B}) \xrightarrow{p} (Y, \mathcal{B}') \tag{39}$$

may be regarded as a probabilistic mapping

$$(X, \mathcal{B}) \xrightarrow{\delta_p} (Y, \mathcal{B}') \tag{40}$$

defined by the one point measure

$$\begin{array}{rcccl} \delta_p(x, \cdot) & : & \mathcal{B}' & \to & [0,1] \\ & : & B' & \mapsto & \begin{cases} 1 & \text{If } p(x) \in B' \\ 0 & \text{If } p(x) \notin B' \end{cases} \end{array} \tag{41}$$

Thus δ_p assigns to x the one-point measure, or dirac measure, on $(Y, \mathcal{B}')$ which is concentrated at $p(x)$. Probabilistic mappings which are one-point measures are called **deterministic mappings**. We use the δ notation to denote all such deterministic mappings, and the subscript to denote the corresponding measurable mapping which corresponds to it.

If $(X, \mathcal{B})$ is a measurable space and $id_X : X \to X$ is the identity map on X as a set, then

$$(X, \mathcal{B}) \xrightarrow{\delta_{id_X}} (X, \mathcal{B}) \tag{42}$$

is the identity mapping on $(X, \mathcal{B})$ in $\mathcal{P}$. With these facts, composition, associativity, and identity, it follows that $\mathcal{P}$ is a category.

2 Some Properties of $\mathcal{P}$

We enumerate some properties of the category of probabilistic mappings required for modeling the recognition problem.

1. If

$$(X, \mathcal{B}) \xrightarrow{\delta_p} (Y, \mathcal{B}') \xrightarrow{\delta_q} (Z, \mathcal{B}'') \tag{43}$$

are deterministic mappings then

$$\delta_{q \circ p} = \delta_q \circ \delta_p. \tag{44}$$

With this result we obtain the subcategory of deterministic mappings $\mathcal{P}_{det}$ which has the same objects as $\mathcal{P}$ but only those arrows which are deterministic. The category $\mathcal{P}_{det}$ is equivalent to the category of measurable spaces, often denoted by $\mathcal{M}eas$ or

$\mathcal{M}es$.

2. A probabilistic mapping

$$1 \xrightarrow{P} (X, \mathcal{B}) \tag{45}$$

where 1 is the one-point space with the only possible algebra defined on it can be construed as a probability measure on $(X, \mathcal{B})$ since P is (isomorphic to) a univariate function,

$$P : \mathcal{B} \to [0, 1].$$

The composition of a probabilistic mappings T with P

$$1 \xrightarrow{P} (X, \mathcal{B}) \xrightarrow{T} (Y, \mathcal{B}') \tag{46}$$

is defined as the induced distribution on $(Y, \mathcal{B}')$. When T is deterministic then classically T is called a "random variable".

Given that $1 \xrightarrow{P} (X, \mathcal{B})$ is just a probability measure on $(X, \mathcal{B})$ it makes sense to call a probabilistic map

$$(X, \mathcal{B}) \xrightarrow{T} (Y, \mathcal{B}') \tag{47}$$

a *variable* probability measure. This concept is one of the generalizations that category theory provides; we no longer need consider just points which are maps whose domain is the terminal object which is denoted by 1. Thus the probabilistic mapping P above is a point, while T is referred to as an (generalized) element. (The older terminology for generalized element is *variable element*.)

3. The category $\mathcal{P}$ has products. This follows from the standard tensor product construction. Given probabilistic mappings μ and ν the product is given by

$$\begin{tikzcd}
 & (T, \mathbb{C}) \arrow[ld, "\mu"'] \arrow[d, "\mu \otimes \nu"] \arrow[rd, "\nu"] & \\
(X_1, \mathbb{B}_1) & (X_1 \times X_2, \mathbb{B}_1 \otimes \mathbb{B}_2) \arrow[l, "\delta_{\pi_1}"] \arrow[r, "\delta_{\pi_2}"'] & (X_2, \mathbb{B}_2)
\end{tikzcd}$$

where $(\mu \otimes \nu)\,(t, (B_1, B_2)) = \mu(t, B_1)\nu(t, B_2)$ and the projections δ_{π_1} and δ_{π_2} are the deterministic probabilistic mappings determined from the canonical projections

$$\begin{tikzcd}
(X_1, \mathbb{B}_1) & (X_1 \times X_2, \mathbb{B}_1 \otimes \mathbb{B}_2) \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & (X_2, \mathbb{B}_2)
\end{tikzcd}$$

where $\mathbb{B}_1 \otimes \mathbb{B}_2$ is the product of $\mathbb{B}_1$ and $\mathbb{B}_2$ in the category of measurable spaces. The commutativity of the biproduct diagram follows from

$$
\begin{aligned}
(\delta_{\pi_1} \circ \mu \otimes \nu)\,(t, B_1) &= \textstyle\int_{X_1 \times X_2} \delta_{\pi_1}((x_1, x_2), B_1) d(\mu \otimes \nu)_t \\
&= \textstyle\int_{X_1 \times X_2} \chi_{B_1 \times X_2} d(\mu \otimes \nu)_t \\
&= (\mu \otimes \nu)_t (B_1 \times X_2) \\
&= \mu(t, B_1)\, \underbrace{\nu(t, X_2)}_{=1}
\end{aligned}
\tag{48}
$$

so $\delta_{\pi_1} \circ (\mu \otimes \nu) = \mu$ and similarly $\delta_{\pi_2} \circ (\mu \otimes \nu) = \nu$.

Arbitrary products are similar using the standard construction for arbitrary products in the category of measurable spaces.

3 Further properties of $\mathcal{P}$

1. The inclusion functor of the subcategory $P_{det} \overset{i}{\hookrightarrow} \mathcal{P}$ has a right adjunct

$$\begin{tikzcd}
\mathcal{P}_{det} \arrow[r, hook, shift left, "i"] & \mathcal{P} \arrow[l, shift left, "\mathcal{D}"]
\end{tikzcd} \qquad i \dashv \mathcal{D}$$

defined by

$$\mathcal{D}(X, \mathcal{B}) = \{1 \xrightarrow{P} (X, \mathcal{B})\}$$

which is the set of all probability measures on $(X, \mathcal{B})$, and endowed with the smallest σ-algebra such that for each $A \in \mathcal{B}$, the evaluation function

$$
\begin{array}{rclcl}
ev(-, A) & : & \mathcal{D}(X) & \to & [0, 1] \\
 & : & P & \mapsto & P(A)
\end{array}
$$

is measurable, when $[0, 1]$ has the Borel σ-algebra.

We denote this σ-algebra by $\mathcal{DB}$.

On arrows, for $(X, \mathcal{B}) \xrightarrow{T} (X', \mathcal{B}')$, define

$$\begin{tikzcd}
(\mathcal{D}X, \mathcal{DB}) \arrow[r, "\mathcal{D}T"] & (\mathcal{D}X', \mathcal{DB}')
\end{tikzcd}$$

by

$$[\mathcal{D}(T)](P,\mathcal{A}) = \begin{cases} 1 & \text{If } T \circ P \in \mathcal{A} \\ 0 & \text{otherwise} \end{cases} .$$

.

The counit of the adjunction,

$$\epsilon : i \circ \mathcal{D} \to Id_{\mathcal{P}}$$

is given at component $(X, \mathcal{B})$ by the *evaluation* probabilistic mapping

$$\begin{tikzcd} i(\mathcal{D}(X,\mathcal{B})) \arrow[r, "ev_{(X,\mathcal{B})}"] & (X,\mathcal{B}) \end{tikzcd}$$

where

$$ev_{(X,\mathcal{B})}(P,B) = P(B).$$

This is a natural transformation because

$$\begin{tikzcd} (X,\mathcal{B}) \arrow[d, "T"] & i(\mathcal{D}(X,\mathcal{B})) \arrow[r, "ev_{(X,\mathcal{B})}"] \arrow[d, "\mathcal{D}T"] & (X,\mathcal{B}) \arrow[d, "T"] \\ (X',\mathcal{B}') & i(\mathcal{D}(X,\mathcal{B})) \arrow[r, "ev_{(X',\mathcal{B}')}"] & (X',\mathcal{B}') \end{tikzcd}$$

commutes since

$$\begin{aligned} (ev_{(X,\mathcal{B})} \circ T)(P,B') &= \int_X \underbrace{T(\cdot,B')}_{limit_{n\to\infty}\left\{\sum_{i=1}^n a_i \chi_{T_{B'}(a_i^\epsilon)}\right\}} d[ev_{(X,\mathcal{B})}(P,\cdot)] \\ &= limit_{n\to\infty}\left\{\sum_{i=1}^n a_i ev(P, T(\cdot,B')^{-1}(a_i^\epsilon))\right\} \\ &= limit_{n\to\infty}\left\{\sum_{i=1}^n a_i P\left(T(\cdot,B')^{-1}(a_i^\epsilon)\right)\right\} \\ &= b_* \qquad \text{some real value} \end{aligned}$$

whereas

$$\begin{aligned}\mathcal{D}\circ ev_{(X',\mathcal{B}')}(P,B') &= \int_{X'} \underbrace{ev_{(X',\mathcal{B}')}(\cdot,B')}_{limit_{n\to\infty}\left\{\sum_{j=1}^m b_j \chi_{ev_{(X',\mathcal{B}')}(\cdot,B')^{-1}(b_j^\epsilon)}\right\}} d[\mathcal{D}T(P,\cdot)] \\ &= limit_{n\to\infty}\left\{\sum_{j=1}^m b_j \mathcal{D}(T)(P, ev_{(X',\mathcal{B}')}(\cdot,B')^{-1}(b_j^\epsilon)\right\}\end{aligned}$$

But

$$\mathcal{D}(T)(P, ev_{X'}(\cdot,B')^{-1}(b_j^\epsilon)) = \begin{cases} 1 & \text{iff } P\circ T \in ev_{X'}(\cdot,B')^{-1}(b_j^\epsilon) \\ 0 & otherwise \end{cases}$$

$$= \begin{cases} 1 \quad \text{iff} & \underbrace{(P\circ T)(B')}_{\begin{aligned} &= \int_{X'} T(\cdot,B')dP \\ &= limit_{n\to\infty}\left\{\sum_{i=1}^n a_i P\left(T(\cdot,B')^{-1}(a_i^\epsilon)\right)\right\} \\ &= b_* \end{aligned}} \in b_j^\epsilon \\ 0 & otherwise \end{cases}$$

Thus at each fixed m, there exist only one nonzero term b_k^m (the index k depends upon the index m) such that $b_* \in [b_k - \epsilon/2, b_k + \epsilon/2)$. Thus, using the fact the ϵ intervals are disjoint,

$$\sum_{j=1}^m b_j \mathcal{D}(T)(P, ev_{X'}(\cdot,B')^{-1}(b_j^\epsilon)) = b_k^m$$

and consequently

$$(\mathcal{D}(T)\circ ev_{X'})(P,B') = limit_{n\to\infty}\{b_k^m\} = b_*$$

2. The coproduct of a collection of objects $\{(X_i,\mathcal{B}_i)\}_{i\in I}$ is given by

$$\begin{tikzcd} (X_i,\mathcal{B}_i) \arrow[r, "\delta_{\iota_i}"] & (\dot\cup_{i\in I}X_i, \langle \dot\cup \mathcal{B}_i\rangle) \end{tikzcd}$$

where ι_i is the inclusion map into the disjoint union of the sets, and $\langle \dot{\cup} \mathcal{B}_i \rangle$ is the σ-algebra generated by the disjoint union of all the component σ-algebras. The inclusion map is a measurable function with respect to this algebra on the set $\dot{\cup}_{i \in I} X_i$. Each such inclusion ι_i has a measurable retraction r_i; choose an element x_{i*} and define

$$\begin{array}{rlll} r_i & : & \dot{\cup}_{i \in I} X_i & \to & X_i \\ & : & y & \mapsto & \begin{cases} y & if \quad y \in X_i \\ x_{i*} & otherwise \end{cases} \end{array}$$

This is measurable since

$$r_i^{-1}(B_i) = \begin{cases} B_i \cup \left(\dot{\bigcup}_{\substack{j \in I \\ j \neq i}} X_j \right) & if x_{i*} \in B_i \\ B_i & otherwise \end{cases}$$

When challenged with a family of measurable mappings $\{\alpha_i\}_{i \in I}$ the unique mapping θ making the diagram

$$\begin{tikzcd}
 & (Y, \mathcal{C}) \\
(X_i, \mathcal{B}_i) \arrow[ur, "\alpha_i"] \arrow[r, "\delta_{\iota_i}"] & (\dot{\cup}_{i \in I} X_i, \langle \dot{\cup} \mathcal{B}_i \rangle) \arrow[u, "\theta"']
\end{tikzcd}$$

commute is given by

$$\theta = \alpha_i \circ \delta_{r_i}.$$

3. $\mathcal{P}$ has coequalizers.

   Consider the two parallel arrows

$$\begin{tikzcd}(X,\mathcal{B}) \arrow[r, shift left, "S"] \arrow[r, shift right, "T"'] & (X',\mathcal{B}')\end{tikzcd}$$

Define

$$\begin{aligned}\mathcal{B}'' &= \{B_i' \in \mathcal{B}' | \forall x \in X \quad S(x, B_i') = T(x, B_i') \text{ and} \\ &\qquad \forall i, j \quad S(x, B_i' \cap B_j') = T(x, B_i' \cap B_j')\}\end{aligned}$$

Lemma. The set $\mathcal{B}''$ is a sub σ-algebra of

$\mathcal{B}'$.

Proof. (i) $\emptyset, X \in \mathcal{B}''$ since $S_x(\emptyset) = 0 = T_x(\emptyset)$ and $S_x(X) = 1 = T_x(X)$. (ii) Suppose $B' \in \mathcal{B}''$ so $S_x(B') = T_x(B')$. Then since

$$\begin{aligned}1 = S_x(X) &= S_x(B') + S_x(B'^c) \\ &= T_x(B') + S_x(B'^c)\end{aligned}$$

from which it follows

$$\begin{aligned}1 - T_x(B') &= S_x(B'^c) \\ T_x(B'^c) &= S_x(B'^c)\end{aligned}$$

hence

$$B'^c \in \mathcal{B}''.$$

(iii) Let $\{B_i' \in \mathcal{B}''\}_{i \in I}$. Then

$$\begin{aligned}S_x(B_i' \cup B_j') &= S_x(B_i') + S_x(B_j') - S_x(B_i' \cap B_j') \\ &= T_x(B_i') + T_x(B_j') - T_x(B_i' \cap B_j') \\ &= T_x(B_i' \cup B_j')\end{aligned}$$

By iteration, for any finite integer n, $S_x(\cup_{i=1}^n B_i') = T_x(\cup_{i=1}^n B_i')$. Since $S_x(\cup_{i \in I} B_i')$ is bounded from above,

$$\begin{aligned}S_x(\cup_{i \in I} B_i') &= limit_{n \to \infty}\{S_x(\cup_{i=1}^n B_i')\} \\ &= limit_{n \to \infty}\{T_x(\cup_{i=1}^n B_i')\} \\ &= T_x(\cup_{i \in I} B_i')\end{aligned}$$

qed.

We can now show that the coequalizer of the parallel pair S and T is given by the deterministic mapping $\delta_{id_{X'}}$

$$\begin{tikzcd}
(X,\mathcal{B}) \arrow[r, shift left, "S"] \arrow[r, shift right, "T"'] & (X',\mathcal{B}') \arrow[r, "\delta_{id_{X'}}"] & (X',\mathcal{B}'')
\end{tikzcd}$$

which is the restriction to sub σ-algebra $\mathcal{B}''$.

Given any other probabilistic mapping U satisfying $U \circ S = U \circ T$ we must show there exist a unique mapping θ making the diagram

$$\begin{tikzcd}
 & & (Y,\mathcal{C}) \\
(X,\mathcal{B}) \arrow[r, shift left, "S"] \arrow[r, shift right, "T"'] & (X',\mathcal{B}') \arrow[r, "\delta_{id_{X'}}"] \arrow[ur, "U"] & (X',\mathcal{B}'') \arrow[u, "\theta"']
\end{tikzcd}$$

commute. Define

$$\theta(x,C) = U(x,C).$$

From this it follows that

$$\begin{aligned}
(\theta \circ \delta_{id})\,(x,C) &= \int_{X'} \underbrace{\theta(\cdot,C)}_{limit_{n\to\infty}\{\sum_{i=1}^{n} a_i \chi_{\theta_C^{-1}(a_i^\epsilon)}\}} d\delta_{id} \\
&= limit_{n\to\infty}\{\textstyle\sum_{i=1}^{n} a_i \underbrace{\delta_{id}(x,\theta(\cdot,C)^{-1}a_i^\epsilon)}_{=\begin{cases} 1 & \text{iff } \theta(x,C) \in a_i^\epsilon \\ 0 & \text{otherwise} \end{cases}} \} \\
&= \theta(x,C) \\
&= U(x,C)
\end{aligned}$$

This proves existence. Uniqueness employs the same argument; simply replace θ with another mapping, say η, making the diagram commute and one concludes that $\eta = U = \theta$.

4. By the previous two results, $\mathcal{P}$ has all colimits.

5. The object 1 is a seperator in $\mathcal{P}$.

   Proof. Suppose the two parallel arrows

$$\begin{tikzcd}(X, \mathbb{B}) \arrow[r, shift left, "T"] \arrow[r, shift right, "R"'] & (Y, \mathbb{C})\end{tikzcd}$$

   are distinct, say

$$T(x, C) = a_* \neq b_* = R(x, C).$$

   Then the one point measure

$$\begin{tikzcd}1 \arrow[r, "\delta_x"] & (X, \mathbb{B})\end{tikzcd}$$

   seperates the pair since $(T \circ \delta_x)(C) = a_*$ while $(R \circ \delta_x)(C) = b_*$.

6. The category $\mathcal{P}$ is concrete.

   Proof. We must prove that there exist a faithful functor

   $\mathcal{P} \to \mathcal{S}et$.

   We have the adjunction

$$\begin{tikzcd}\mathcal{S}et \arrow[r, shift left, "Dis"] & \mathcal{P}_{det} \arrow[l, shift left, "U"]\end{tikzcd} \qquad\qquad Dis \dashv U$$

where Dis assigns the discrete σ-algebra to a set X and U is the "forgetful" functor defined on the objects of the deterministic subcategory $\mathcal{P}_{det}$ by $U((X,\mathcal{C})) = X$ and on the arrows by $U(\delta_f) = f$. The unit of the adjunction is the identity arrow and the existence and uniqueness requirements given an arbitrary (set) function $f : X \to Y$ follows from the commutative diagram

$$\begin{tikzcd}
X \arrow[r, "id_X"] \arrow[dr, "f"'] & U((X,\mathcal{P}X)) \arrow[d, "U(\delta_f)"] & & (X,\mathcal{P}X) \arrow[d, "\delta_f"] \\
& U((Y,\mathcal{B})) & & (Y,\mathcal{B})
\end{tikzcd}$$

which proves that the identity mapping is a universal arrow from the object X to the functor U. Thus the unit of the adjunction, $\eta : Id \to U \circ Dis$ is defined componentwise by $\eta_X = id_X$. Consider the pair of adjunctions

$$\begin{tikzcd}
\mathcal{S}et \arrow[r, shift left, "Dis"] & \mathcal{P}_{det} \arrow[l, shift left, "U"] \arrow[r, shift left, "i"] & \mathcal{P} \arrow[l, shift left, "\mathcal{D}"]
\end{tikzcd} \qquad i \circ Dis \dashv U \circ \mathcal{D}$$

Both the functors $\mathcal{D}$ and U are faithful; hence so is their composite.

4 Probability Measures and the Part/Whole Relation

The objects we need to consider can be viewed as a collection of point features. Previously we have considered an object to consist of k-points in $\mathbb{R}^3$. What we really require is that the object be considered as a collection of N parts, with each part consisting of k_i points in $\mathbb{R}^3$.

Let M stand for *model*; formally, it is just a set $M \subset \mathbb{R}^3$. It consists of a collection of parts

$$\begin{tikzcd}
m_j \arrow[r, hook, "\iota_j"] & M
\end{tikzcd}$$

where ι_j is set inclusion, and the sets m_i represent points, lines, surfaces, etc. Endow M with the σ-algebra generated by $\{B \cap M | B \in \mathcal{B}_{\mathbb{R}^3}\}$ and its' collection of parts $\{m_j\}_{j=1}^N$, where $\mathcal{B}_{\mathbb{R}^3}$ is the Borel σ-algebra on $\mathbb{R}^3$. Denote this σ-algebra by $\mathcal{B}_M$. Thus $(M, \mathcal{B}_M)$ is a measurable space and hence an object of $\mathcal{P}$.

Because the parts m_j are measurable sets it follows that the sub σ-algebra

$$\mathcal{B}_j = \{B \cap m_j | B \in \mathcal{B}_M\} \tag{49}$$

is a σ-algebra on m_j, and so each $(m_j, \mathcal{B}_j)$ is an object of $\mathcal{P}$, and each inclusion map is a measurable map, and hence a deterministic mapping in $\mathcal{P}$

$$\begin{tikzcd}(m_j, \mathcal{B}_j) \arrow[r, hook, "\delta_{\iota_j}"] & (M, \mathcal{B}_M).\end{tikzcd}$$

If P_1 and P_2 are probability measures on these parts

$$\begin{tikzcd} & (m_1, \mathcal{B}_1) \arrow[dr, "\delta_{\iota_1}"] & \\ 1 \arrow[ur, "P_1"] \arrow[dr, "P_2"'] & & (M, \mathcal{B}) \\ & (m_2, \mathcal{B}_2) \arrow[ur, "\delta_{\iota_2}"'] & \end{tikzcd}$$

then both $\delta_{\iota_1} \circ P_1$ and $\delta_{\iota_2} \circ P_2$ are probability measures on M. Furthermore any convex

combination of these two probability measures on M is also a

probability measure on M

$$\begin{tikzcd}1 \arrow[rr, "\theta_1(\delta_{\iota_1} \circ P_1) + \theta_2(\delta_{\iota_2} \circ P_2)"] & & (M, \mathcal{B})\end{tikzcd} \qquad \theta_1 + \theta_2 = 1, \quad \theta_i \geq 0.$$

The scalars θ_i's are the "weights" on the respective parts. Such a convex combination of probability measures on the "whole object" M in terms of its' parts m_i is very appealing for the recognition problem. The individual probabilities on the parts m_i can be calculated as in the applications section, say by the Monte Carlo method as mentioned, and then the probability on the whole can be obtained as

$$\begin{tikzcd} 1 \arrow[r, "\sum_{i=1}^{N}(\delta_{\iota_i} \circ P_i)"] & (M,\mathcal{B}) \end{tikzcd} \qquad \sum_{i=1}^{N} \theta_i = 1, \quad \theta_i \geq 0.$$

The challenge resides is in choosing the weights θ_i in a reasonable fashion.

5 Sheaf-Theoretic Aspects

A convex combination of probability measures P_1 and P_2 as in the preceding expression does not take into account the fact that m_1 and m_2 may intersect. To construct amalgamations which take into account intersections it is necessary to embed $\mathcal{P}$ into a slightly larger category which allows *scalar* endomappings defined as

$$(X,\mathcal{B}) \xrightarrow{\delta_{id}^c} (X,\mathcal{B})$$

defined by

$$\begin{array}{rcccl} \delta_{id}^c & : & X \times \mathcal{B} & \to & [0,\infty) \\ & : & (x,B) & \mapsto & \begin{cases} c & \text{if } x \in B \\ 0 & \text{otherwise} \end{cases} \end{array}$$

As the notation suggest these scalar mappings are deterministic, induced by the identity maps, but of total measure c. It is convenient to consider probability measures defined on the coproducts

$$\begin{tikzcd} 1 \arrow[r, "P_i"] & 1 + (m_i,\mathcal{B}_i) \end{tikzcd}$$

where $P(1) = 0$.

Consider the diagram

$$\begin{tikzcd}
 & 1+m_1 \arrow[dr, "\delta_{\iota_1}"] & \\
1 \arrow[ur, "P_1"] \arrow[dr, "P_2"'] & & 1+M \\
 & 1+m_2 \arrow[ur, "\delta_{\iota_2}"'] &
\end{tikzcd}$$

where we have dropped explicit mention of the σ-algebras and where we have redefined ι to account for the new element

$* \in 1 + m_j$,

$$\iota_j(x) = \begin{cases} x & \forall x \in m_j \\ * & x = * \end{cases}.$$

Each measurable mapping ι_j has a retraction s_j

$$\begin{tikzcd}
1+m_j \arrow[r, shift left, "\iota_j"] & 1+M \arrow[l, shift left, "s_j"]
\end{tikzcd} \qquad s_j \circ \iota_j = id_{1+m_j}$$

which is measurable and defined by

$$s_j(x) = \begin{cases} x & \text{If } x \in m_j \\ * & \text{otherwise} \end{cases}.$$

In the category of measurable spaces, let

$$\begin{tikzcd}
 & 1+m_1 \arrow[dr, "\iota_1"] & \\
1+(m_1 \cap m_2) \arrow[ur, "p_1"] \arrow[dr, "p_2"'] & & 1+M \\
 & 1+m_2 \arrow[ur, "\iota_2"'] &
\end{tikzcd}$$

be the pullback of ι_1 and ι_2. Then the inclusion maps $1+(m_1 \cap m_2) \hookrightarrow 1+m_j$ both have a measurable retraction r_j defined by

$$r_j(x) = \begin{cases} x & \text{If } x \in m_1 \cap m_2 \\ * & \quad \text{otherwise} \end{cases} .$$

In $\mathcal{P}$, consider the diagram

$$\begin{tikzcd}
 & & & 1+m_1 \arrow[dl, shift left, "\delta_{r_1}"] \arrow[dr, shift left, "\delta_{\iota_1}"] & \\
1 \arrow[urrr, "P_1"] \arrow[drrr, "P_2"'] & & 1+(m_1 \cap m_2) \arrow[ur, shift left, "\delta_{p_1}"] \arrow[dr, shift left, "\delta_{p_2}"'] & & 1+M \arrow[ul, shift left, "\delta_{s_1}"] \arrow[dl, shift left, "\delta_{s_2}"'] \\
 & & & 1+m_2 \arrow[ul, shift left, "\delta_{r_2}"'] \arrow[ur, shift left, "\delta_{\iota_2}"'] &
\end{tikzcd}$$

Suppose P_1 and P_2 agree on $m_1 \cap m_2$,

$$P_{12} \stackrel{\triangle}{=} \delta_{r_1} \circ P_1 = \delta_{r_2} \circ P_2.$$

Define (the nonprobabilistic mapping)

$$\hat{P} = \delta_{\iota_1} \circ P_1 + \delta_{\iota_2} \circ P_2 - \delta_{\iota_2 \circ p_2 \circ r_2} \circ P_2$$

Then the mapping

$$P \stackrel{\triangle}{=} \delta_{id}^{\frac{1}{\hat{P}(1+M)}} \circ \hat{P} : 1 \to 1+M.$$

is a probabilistic mapping which is the amalgamation of P_1 and P_2. It is of measure one because of the scaling; it satisfies finite additivity because each component term does.

The restriction of P to $1+m_1$, defined as $P|_{1+m_1} \stackrel{\triangle}{=} \delta_{s_1} \circ \delta_{id}^{\hat{P}(1+M)} \circ P$, gives

$$\begin{aligned} P|_{1+m_1} &= \underbrace{\delta_{s_1 \circ \iota_1}}_{=id} \circ P_1 + \underbrace{\delta_{s_1 \circ \iota_2} \circ P_2 - \delta_{s_1 \circ \iota_2 \circ p_2 \circ r_2} \circ P_2}_{=0} \\ &= P_1 \end{aligned}$$

where the latter two terms cancel each other out. This can be verified by looking at these terms on the generating elements of the σ-algebra of $1+M$ and the subalgebras:

Consider the event $B \cap m_1$, where $B \in \mathcal{B}_M$,

$$\begin{array}{rcl}
(\delta_{s_1 \circ \iota_2} \circ P_2)\,(B \cap m_1) & = & \int_{1+m_2} \delta_{s_1 \circ \iota_2}(\cdot, B \cap m_1) dP_2 \\
& = & \int_1 \delta_{s_1 \circ \iota_2}(\cdot, B \cap m_1) dP_2 + \int_{m_1 \cap m_2} \delta_{s_1 \circ \iota_2}(\cdot, B \cap m_1) dP_2 \\
& & + \int_{m_1^c} \delta_{s_1 \circ \iota_2}(\cdot, B \cap m_1) dP_2 \\
& = & 0 + P_2(B \cap m_1 \cap m_2) + 0
\end{array}$$

where, as usual, for any measurable $Z \subset 1+m_1$, $\int_Z f(\cdot) dP_1 \triangleq \int_{1+m_1} \chi_Z f(\cdot) dP_1$.

Decomposing the integral for $(\delta_{s_1 \circ \iota_2 \circ p_2 \circ s_2} \circ P_2)(B)$ into three components shows it is equal to $P_2(B \cap m_1 \cap m_2)$.

This same procedure works for any finite number of parts $\{m_i\}_{i=1}^n$ giving an amalgamation with the desired restriction properties. The more familiar functorial view is given by employing the composite functor

$$\begin{array}{cccccc}
\mathcal{UD} & : & \mathcal{P} & \to & \mathcal{S}et \\
 & :_{ob} & (X, \mathcal{B}) & \mapsto & \mathcal{D}X \\
 & :_{ar} & (X, \mathcal{B}) \xrightarrow{T} (X', \mathcal{B}') & \mapsto & \mathcal{D}X \xrightarrow{\mathcal{D}(T)} \mathcal{D}Y
\end{array}$$

where

$$[\mathcal{D}(T)](P) = T \circ P$$

Then given a finite cover $\{m_i\}_{i=1}^n$ of M, and a matching compatible family of elements $\{P_i \in (\mathcal{UD})(m_i)\}_{i=1}^n$ there exist a unique amalgamation $P \in \mathcal{UD}(M)$ (constructed as above) satisfying the restriction property.

6 Gluing and Restrictions of Probabilities.

Suppose we are working in a topos $\mathcal{C}$ with a subobject classifier Ω. Consider the diagram

$$\begin{tikzcd}
A_1 \arrow[dr, "f_1"] & \\
& X \\
A_2 \arrow[ur, "f_2"'] &
\end{tikzcd}$$

where f_1 and f_2 are parts of X (monics). Taking the pullback diagram gives

$$\begin{tikzcd}
& A_1 \arrow[dr, "f_1"] & \\
A_1 \cap A_2 \arrow[ur, "p_1"'] \arrow[dr, "p_2"] & & X \\
& A_2 \arrow[ur, "f_2"'] &
\end{tikzcd}$$

There are two covariant functors of interest: The functor

$$\begin{array}{rlll}
\mathcal{G} & : \ \mathcal{C} & \rightarrow & Set \\
& :_{ob} \ A & \rightarrow & \{\Omega^A \xrightarrow{P} [0,1] : P \text{ a probability measure}\} \\
& :_{ar} \ A \xrightarrow{f} B & \mapsto & \{\Omega^A \xrightarrow{P} [0,1] \text{ :P a p.m.}\} \xrightarrow{\Omega^f} (\{\Omega^B \xrightarrow{P} [0,1] \text{ :P a p.m.}\}
\end{array} \tag{50}$$

where $\Omega^{\cdot}$ is the power object functor (contravariant hom functor $hom(\cdot, \Omega)$) mapping the objects and arrows into the category of sets, where for $B \xrightarrow{t} \Omega$, with transpose (adjunct) $1 \xrightarrow{\bar{t}} \Omega^B$, $\Omega^f(P \circ \bar{t}) = P \circ \overline{t \circ f}$.

The second functor of interest is the contravariant functor

$$\begin{array}{rlll}
\mathcal{F} & : \ \mathcal{C} & \rightarrow & Set \\
& :_{ob} \ A & \mapsto & \{\Omega^A \xrightarrow{P} [0,1] : P \text{ a probability measure}\} \\
& : \ P_A & \mapsto & P_A \circ im_f
\end{array} \tag{51}$$

where $im.$ is the direct image functor. (The image functor goes by many names. Another common symbol is $\exists.$.) Both functors $\mathcal{F}$ and $\mathcal{G}$ have the same mapping on objects. Because of the relationship between the covariant imaging functor $im.$ and the contravariant power object functor $\Omega^{\cdot}$ we obtain the diagram

$$\begin{tikzcd}
 & \Omega^{A_1} \arrow[dl, shift left, "\Omega^{p_1}"'] \arrow[dr, shift left, "im_{f_1}"'] \arrow[drr, "P_1"] & & \\
\Omega^{A_1 \cap A_2} \arrow[ur, shift left, "im_{p_1}"'] \arrow[dr, shift left, "im_{p_2}"] & & \Omega^X \arrow[ul, shift left, "\Omega^{f_1}"'] \arrow[dl, shift left, "\Omega^{f_2}"] \arrow[r, "P_{\mathbf{X}}"] & {[0,1]} \\
 & \Omega^{A_2} \arrow[ul, shift left, "\Omega^{p_2}"] \arrow[ur, shift left, "im_{f_2}"] \arrow[urr, "P_2"'] & &
\end{tikzcd}$$

The Beck-Chevalley Theorem says that

$$\Omega^{f_2} \circ im_{f_1} = im_{p_2} \circ \Omega^{p_1} \tag{52}$$

It is this "commutativity condition" which gives rise to the sheaf theoretic aspects which we now describe.

Suppose P_1 and P_2 are two separate probability measures on A_1 and A_2 respectively. Hence they are maps

$$\begin{array}{rcrcl} P_1 & : & \Omega^{A_1} & \to & [0,1] \\ P_2 & : & \Omega^{A_2} & \to & [0,1] \end{array} \tag{53}$$

(More generally P_1 and P_2 can be any measures on any boolean subalgebra of the power objects Ω^{A_1} and Ω^{A_2} respectively.) If these two measures P_1 and P_2 agree on $A_1 \cap A_2$ then

$$P_1 \circ im_{p_1} = P_2 \circ im_{p_2}. \tag{54}$$

Both of these quantities are the restriction maps - the restriction of P_i on A_i to $A_1 \cap A_2$ for $i = 1, 2$. From the above diagram we obtain the desired amalgamation of P_X of P_1 and P_2 as

$$P_X = P_1 \circ \Omega^{f_1} + P_2 \circ \Omega^{f_2} - P_2 \circ im_{p_2} \circ \Omega^{f_1 \cap f_2}. \tag{55}$$

Because f_1 is a part of X, the restriction of P_X to A_1 is

$$P_X \circ im_{f_1} = P_1 \circ \Omega^{f_1} \circ im_{f_1} + P_2 \circ \Omega^{f_2} \circ im_{f_1} - P_2 \circ im_{p_2} \circ \Omega^{f_1 \cap f_2} \circ im_{f_1}. \quad (56)$$

Using the three properties

$$\begin{aligned} &\Omega^{f_i} \circ im_{f_i} = id_{A_i} \\ &\Omega^{f_1 \cap f_2} = \Omega^{p_1} \circ \Omega^{f_1} = \Omega^{p_2} \circ \Omega^{f_2} \\ &P_1 \circ im_{p_1} = P_2 \circ im_{p_2} \quad \text{Matching Condition} \end{aligned} \quad (57)$$

along with the Beck-Chevalley Condition

$$\Omega^{f_2} \circ im_{f_1} = im_{p_2} \circ \Omega^{p_1} \quad (58)$$

reduces the above equation to

$$\begin{aligned} P_X \circ im_{f_1} &= P_1 + P_1 \circ im_{p_1} \circ \Omega^{p_1} - P_1 \circ im_{p_1} \circ \Omega^{p_1} \\ &= P_1 \end{aligned} \quad (59)$$

Similarly because f_2 is a part of X it follows that

$$P_X \circ im_{f_2} = P_2. \quad (60)$$

which proves the amalgamations restrict to the parts of which it is composed. This proof extends by induction to an arbitrary covering of parts $\{f_i\}_{i=1}^n$. The restriction that the covers be parts can be removed by using the image of each cover f_i. Consequently there is no loss in generality by assuming the cover to consist of monic maps f_i.

Conversely, given a cover $\{f_i\}_{i=1}^n$ and a measure P_X on X, the restriction of P_X to A_i is given by

$$P_i = P_x \circ im_{f_i} \quad (61)$$

which is a measure on A_i because P_X is a measure. When the maps f_i are monic then im_{f_i} acts as the inclusion map. The quantity P_i can be made into a probability measure by the appropriate scaling.

6.1 Summary

The point to surface matching algorithm is a practical approach to solving the ATR problem because unlike its' predecessor, using point to point matching of "features", it circumvents the combinatorial explosion that occurs with such methods. The representation of objects by CAD models is seemingly obvious, yet the representation of objects by a collection of points is still used in some quarters. This is an unnecessary complication and makes the feasibility of such approaches impractical for real-time computations. The feasibility of the algorithm used to solve the point to surface model resides in the fact that we can "collapse" the objective function - which is none other than the weighted sum of every possibility with respect to the discretization - by exploiting the bilineararity of the inner product. Because the constraints associated with this model are all linear and decoupled the computations of the optimization algorithm (search direction and line search) can be carried out in parallel for each of the unknown parameters. With hindsight this approach seems elementary and obvious. Unlike other ad hoc approaches (such as the Iterated Closest Point Algorithm) it has a theoretical basis behind it. Indeed our derivation of it came from studying the deterministic subcategory of the category of probabilistic mappings which provides a solid foundation for the approach as well as any future studies/extensions of this approach. Using the category of probabilistic mappings a sheaf-theoretic view of the problem can be taken but is not necessary from the computational point of view.

The computational time of the algorithm depends upon the number of image points and the number of surfaces modeling the object ("the resolution" of the CAD model). On a Mac Quad Pro with 4 processors the computation time for our standard test case - the tank with 206 surfaces and the point cloud consisting of 300 image points - is about 2 minutes. By using GPCA this can be reduced an order of magnitude.

The results of the output of the algorithm for classification purposes are, like any other algorithm, heavily dependent upon the data input to the algorithm.

Namely the image data (point cloud of data) must cover a sufficient number of surfaces which “distinquish” the true object from any decoys. Given data from a side view or directly overhead is an extreme case in which it is very difficult to make proper classification because the data falls on surfaces which all lie in hyperplanes which are similar, i.e., have similar surface normals). On the other hand, data taken from an aircraft ladar at $30 - 60$ degrees tends to be sufficient provided occlusion does not block the view of many surfaces defining the object. This algorithm is being used in a SBIR, in joint effort with Etegent Technologies, to further its development and integrate it with the preprocessing of the raw image data to produce an overall ATR system.

A copy of the Mathematica code is included with this report. It is modular and documented sufficiently to be able to follow and use for experimental purposes.

7 Financial Report

The Table below shows the expenditures on a running monthly basis. The graph following it shows represents the block funding on a yearly basis, the green graph is the estimated monthly expenditures, and the red graph is the actual monthly expenditures. The estimated expenditures were based upon monthly linear spending of available funding. Note we did not start charging until March 2005 so the first billing is in the 6th month of the first fiscal year.

There were no travel expenses or material charges; all charges were labor expenses.

Table 1. Quarterly Program-to-Date Expenses

Quarter	Period	Expenditures
1	March 11-June 05	31,555.95
2	July-Sept. 05	30,044.82
3	Oct.-Dec. 05	29,889.26
4	Jan.-March 06	30,489.27
5	April-June 06	31,022.61
6	July-Sept. 06	31,644.84
7	Oct.-Dec. 06	40,461.50
8	Jan.-March 07	42,072.62
9	April-June 07	42,822.62
10	July-Sept. 07	41,283.72
11	Oct.-Dec. 07	40,767.06
12	Jan.-Mar. 08	40,750.00
13	April-June 08	41,000.00
14	July-Sept. 08	44,500.00
15	Oct.-Dec. 08	41,300.00
16	Jan.-March 09	36,000.00
17	April-June 09	33,600.00
18	July 09-Sept. 09	37,400.00
19	Oct. -Dec. 09	20,800.00
20	Jan. 2010- March 2010	33,150.00
21	April -Jun 10	40,551.00
22	July - Sept. 2010	58,200.00
23	October 2010	14328
Totals	March 11, 2005-Oct. 2010	822,695.14

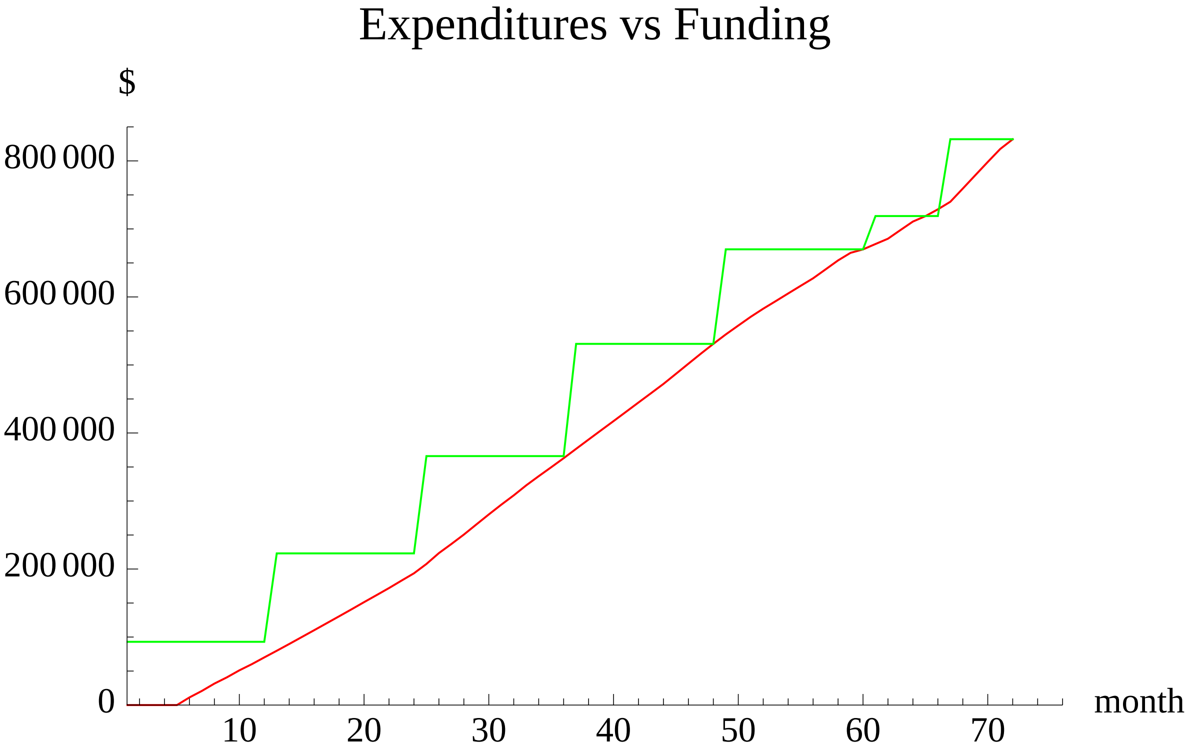


Figure 30: Funding - Expenditure Profile